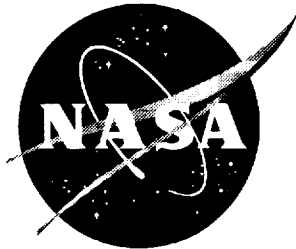


NASA/TM-1999-209530



High Speed Civil Transport Aircraft Simulation: Reference-H Cycle 1

MATLAB Implementation

Robert A. Sotack

*George Washington University, Joint Institute for Advancement of Flight Sciences
Langley Research Center, Hampton, Virginia*

Rajiv S. Chowdhry

Lockheed Martin Engineering & Sciences Company, Hampton, Virginia

Carey S. Buttrill

Langley Research Center, Hampton, Virginia

December 1999

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

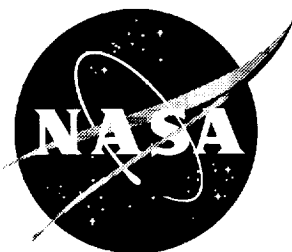
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- Email your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Telephone the NASA STI Help Desk at (301) 621-0390
- Write to:
NASA STI Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/TM-1999-209530



High Speed Civil Transport Aircraft Simulation: Reference-H Cycle 1

MATLAB Implementation

Robert A. Sotack

*George Washington University, Joint Institute for Advancement of Flight Sciences
Langley Research Center, Hampton, Virginia*

Rajiv S. Chowdhry

Lockheed Martin Engineering & Sciences Company, Hampton, Virginia

Carey S. Buttrill

Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

December 1999

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Table of Contents

1	Introduction	1
2	Aircraft Equations of Motion.....	2
2.1	Underlying Assumptions.....	2
2.2	Translational Equations of Motion	2
2.3	Rotational Equations of Motion	4
2.4	Combined Accelerations.....	5
2.5	Explicit Acceleration Solution	6
2.6	Aircraft Mass Center Position	9
2.7	Aircraft Orientation Using Euler Angles.....	9
2.8	Flight Path Calculations	10
2.9	Summary.....	10
3	Aerodynamic Model.....	11
3.1	Aerodynamic Model Description	11
3.2	Control Surfaces	11
3.3	Aerodynamic Force and Moment Calculations	11
3.4	Steady Flow Outputs	14
3.5	Unsteady Flow Outputs	16
3.6	Aerodynamic Model Implementation	18
4	Engine Model.....	21
4.1	Engine Model Description.....	21
4.2	Engine Model Dynamics	21
4.3	Engine Model Buildup Equations	21
4.4	Engine Model Implementation.....	21
5	Mass Model.....	24
5.1	Mass Model Description.....	24
5.2	Mass Model Implementation.....	24
6	Flight Control System Model	26
6.1	Accelerometers	26
6.2	Actuators	26
6.3	Flight Control System.....	26
7	Atmosphere Model	27
7.1	1962 Standard Atmosphere Model.....	27
7.2	Dryden Turbulence Model	27
8	Simulation Structure	29
8.1	Simulation Layout.....	29
8.2	Simulation Implementation.....	29
8.3	Simulation Initialization And Setup	33
8.4	Simulation Results	35
9	Trim Solutions.....	36
9.1	Finding A Trim Solution	36
9.2	Trim Solution Implementation.....	36
10	Linear Models	41
10.1	Introduction	41
10.2	States, Inputs, and Outputs	41
10.3	State Vector and Output Vector Transformations	41
10.4	State-Space Form.....	45
10.5	Units	46
10.6	Linear Model Implementation	49
10.7	Linear Model Results.....	52
11	Concluding Remarks	58

References.....	59
Symbols and Abbreviations.....	60
Simulation Files	63
Appendix A - Apparent Mass Matrix Elements	64
Appendix B - Dynamic Response Comparison To A Rudder Doublet.....	68

1 Introduction

This report describes the mathematical model and associated code used to simulate a high speed civil transport aircraft - the Boeing Reference H configuration. The simulation was constructed to support advanced control law research by providing an appropriate tool for control system engineering. For this reason, the simulation was implemented in a commercially available and widely used controls engineering software package called MATLAB/SIMULINK®.

To date there have been 3 formal releases of the Reference H simulation math model as shown in the table below:

Cycle 1	September, 1994	(Dawdy et al- June, 1994)
Cycle 2A	March, 1995	(Stephens et al - March, 1995)
Cycle 2B	July, 1995	(Stephens et al - July, 1995)

This report describes SIMULINK implementation of the Cycle 1 Reference H simulation model. Future versions of this report will be issued on a regular basis to document subsequent releases of the simulation math model.

MATLAB and its toolboxes provide the linear algebra, matrix computation, numerical analysis capabilities, and advanced control law analysis and synthesis methods that are suitable for control system engineering. SIMULINK provides a graphical, block diagram format for constructing and simulating dynamic systems. The simulation software developed on the MATLAB/SIMULINK platform can be used for obtaining nonlinear dynamic responses of the aircraft and also provides the capability to find trim solutions and linear models at specified flight conditions. Thus, the developed simulation provides the necessary basis for developing and testing candidate flight control systems for a representative high speed civil transport aircraft.

The simulation math model includes the nonlinear, six degree-of-freedom equations that govern the motion of a rigid aircraft in atmospheric flight over a rotating spherical Earth. A Dryden turbulence model and the 1962 Standard Atmosphere model are provided to simulate the Earth atmosphere. The Reference H configuration is modeled using the Boeing cycle 1 simulation data base and consists of a tabular aerodynamic model, an engine model, and a mass model. The aerodynamic model has the option to include the effects of quasi-static elastic deformation on the six rigid-body aerodynamic forces and moments. Landing gear, flight control and the actuator models are not documented in this report will be covered in future releases of the simulation and its documentation.

The equations of motion used in the simulation are presented in Chapter 2. Chapters 3, 4, and 5 present brief descriptions of the Reference H configuration aerodynamic, engine, and mass models used in the simulation. The equations used to compute the accelerations at a given sensor location are presented in Chapter 6. The atmosphere model and the turbulence model implemented in the simulation are described in Chapter 7. Chapter 8 describes the overall simulation layout and the procedure to run the nonlinear simulation with an example case. The trimming procedure and linear model generation procedures with examples are presented in Chapters 9 and 10, respectively.

To validate the developed software, a dynamic check case was generated using the MATLAB/SIMULINK based simulation and compared to dynamic responses with those generated from independent simulations implemented by Boeing Commercial Aircraft Company and by the real-time simulation facility support group at NASA Langley Research Center. The results of the comparison are also included.

® MATLAB and SIMULINK are registered trademarks of The MathWorks, Inc.

2 Aircraft Equations of Motion

2.1 Underlying Assumptions

The equations governing the motion of a rigid aircraft in atmospheric flight are available from several sources (Etkin - 1972 and McFarland - 1975), and are not derived in this paper. The equations discussed in this paper are taken from Etkin - 1972. The equations utilize a body reference frame with the origin at the aircraft center of gravity, schematically shown in Figure 2.1. The positive x axis passes through the aircraft nose. The positive y axis passes through the right wing and the positive z axis passes through the bottom of the aircraft fuselage. The equations of motion were developed based on the following assumptions:

1. The Earth is a sphere rotating with constant rotation ω_E on an axis fixed in inertial space.
2. Acceleration due to gravity is a radial vector which acts at the aircraft mass center.
3. The aircraft is a rigid body which is symmetrical about the plane formed by the body frame x axis and z axis.

2.2 Translational Equations of Motion

Let the translational inertial velocity of an aircraft, expressed in body frame components, be given by $[u \ v \ w]'$ and let the inertial rotational velocity of an aircraft in body frame components be given by $[p \ q \ r]'$. The rotational velocity of the Earth in body frame components is given by

$$\begin{Bmatrix} p_b^E \\ q_b^E \\ r_b^E \end{Bmatrix} = [T_{BV}] \begin{Bmatrix} \cos \lambda \\ 0 \\ -\sin \lambda \end{Bmatrix} \omega_E, \quad (2.1)$$

where λ is aircraft latitude and $[T_{BV}]$ is the transformation matrix from a vehicle-carried local horizontal reference frame to a body reference frame, which is given by

$$[T_{BV}] = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi & \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\ -\cos \phi \sin \psi & +\cos \phi \cos \psi & \\ \cos \phi \sin \theta \cos \psi & \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \\ +\sin \phi \sin \psi & -\sin \phi \cos \psi & \end{bmatrix}, \quad (2.2)$$

where ϕ , θ , and ψ are the Euler roll angle, pitch angle, and yaw angle respectively. The vehicle-carried local horizontal reference frame is attached to the aircraft with the origin at the center of gravity. The positive x axis points to the north, the positive y axis points to the east, and the

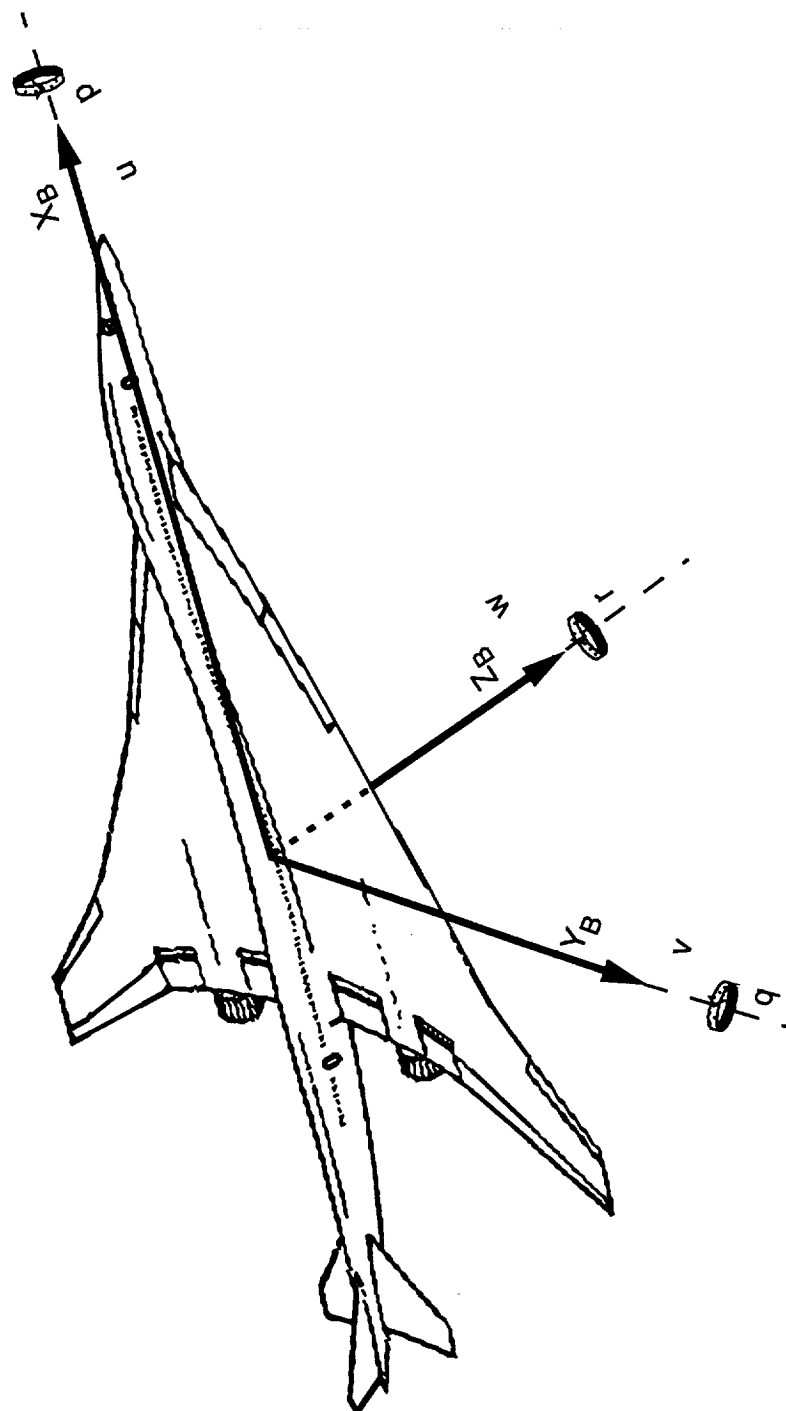


Figure 2.1 The body axis system.

positive z axis points vertically downward toward the Earth's center. Let m be the mass of the aircraft and let $[F_X F_Y F_Z]'$ represent the total external force components acting on the aircraft. The equations describing translational motion are given by

$$m \begin{Bmatrix} \dot{u} + (q + q_b^E)w - (r + r_b^E)v \\ \dot{v} + (r + r_b^E)u - (p + p_b^E)w \\ \dot{w} + (p + p_b^E)v - (q + q_b^E)u \end{Bmatrix} = \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}, \quad (2.3)$$

where the total force vector is given by

$$\begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix} = \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{Aero} + \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{Prop} + \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{Grav}. \quad (2.4)$$

The subscripts 'Aero', 'Prop', and 'Grav' denote aerodynamic, propulsive, and gravitational forces respectively. The components of the gravitational force vector are

$$\begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{Grav} = mg \begin{Bmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{Bmatrix}. \quad (2.5)$$

Gravitational acceleration is denoted by g and is given by

$$g = g_0 (R_E/R)^2, \quad (2.6)$$

where g_0 is the gravitational acceleration at sea level, R_E is the radius of the Earth, and R is the distance from the center of the Earth to the aircraft mass center. Equation (2.3) can be rewritten as

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} + \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} 0 & -(r + r_b^E) & (q + q_b^E) \\ (r + r_b^E) & 0 & -(p + p_b^E) \\ -(q + q_b^E) & (p + p_b^E) & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}. \quad (2.7)$$

2.3 Rotational Equations of Motion

Let the total external moments, in body frame components, acting on the aircraft be given by $[M_X M_Y M_Z]'$. I_x is the roll moment of inertia, I_y is the pitch moment of inertia, I_z is the yaw moment of inertia and I_{xz} is the product of inertia. The equations describing the rotational motion of an aircraft are given by

$$\begin{Bmatrix} \dot{p}I_x - (qp + \dot{r})I_{xz} + qr(I_z - I_y) \\ \dot{q}I_y + (p^2 - r^2)I_{xz} + pr(I_x - I_z) \\ \dot{r}I_z + (qr - \dot{p})I_{xz} + pq(I_y - I_x) \end{Bmatrix} = \begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix}, \quad (2.8)$$

where the total moment components are given by

$$\begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix} = \begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix}_{Aero} + \begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix}_{Prop}. \quad (2.9)$$

Equation (2.8) neglects the rotor terms and the centripetal acceleration associated with the Earth's rotation. The time rate of change of the moments of inertia are assumed to be negligible. However, the moments of inertia can be varied parametrically slowly over time to reflect fuel burn. Equation (2.8) can be rewritten as

$$\begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} \begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = \begin{Bmatrix} M_X \\ M_Y \\ M_Z \end{Bmatrix}. \quad (2.10)$$

2.4 Combined Accelerations

When the body frame inertial velocity components are combined to form the generalized velocity vector given by

$$\{V\} = \begin{Bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{Bmatrix}, \quad (2.11)$$

equation (2.7) and equation (2.10) can be combined to form

$$[M]_I \{\dot{V}\} + [\Omega][M]_I \{V\} = \{F\}. \quad (2.12)$$

The inertial mass matrix $[M]_I$ is given by

$$[M]_I = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & -I_{xz} \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & -I_{xz} & 0 & I_z \end{bmatrix}, \quad (2.13)$$

and $[\Omega]$ represents the body frame inertial rotation, which is given by

$$[\Omega] = \begin{bmatrix} 0 & -(r + r_b^E) & (q + q_b^E) & 0 & 0 & 0 \\ (r + r_b^E) & 0 & -(p + p_b^E) & 0 & 0 & 0 \\ -(q + q_b^E) & (p + p_b^E) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -r & q \\ 0 & 0 & 0 & r & 0 & -p \\ 0 & 0 & 0 & -q & p & 0 \end{bmatrix}, \quad (2.14)$$

and $\{F\}$ is the generalized force vector, given by

$$\{F\} = \begin{Bmatrix} F_X \\ F_Y \\ F_Z \\ M_X \\ M_Y \\ M_Z \end{Bmatrix}. \quad (2.15)$$

2.5 Explicit Acceleration Solution

Like most aerodynamic models, the Reference H aerodynamic model generates forces and moments which contain unsteady flow effects. The unsteady flow contributions to the aerodynamic forces and moments are functions of acceleration. Equation (2.12) must be modified in order to obtain an explicit definition for the aircraft acceleration. Therefore, the force and moment contributions from unsteady flow must be removed from the generalized force vector defined in equation (2.15). This removal is accomplished by separating the aerodynamic forces and moments according to whether the flow is steady or unsteady. To see how this is done, first consider the longitudinal force and moment coefficients generated by the Reference H aerodynamic model. These force and moment coefficients contain linear contributions from non-dimensionalized forms of the time rate of change of angle of attack and the time rate of change of pitch rate. Defining $\dot{\alpha}$ as the time rate of change of angle of attack and \dot{q} as the time rate of change of pitch rate, the longitudinal force and moment coefficients, in stability frame components, are expressed as

$$C_\chi = \dots + C_{\chi_{\ddot{\alpha}}} \left(\frac{\ddot{\alpha} \bar{c}}{2V_T} \right) + C_{\chi_{\dot{\alpha}}} \dot{\alpha} \left(\frac{\bar{c}}{2V_T} \right)^2, \quad (2.16)$$

where ' χ ' can be replaced by ' D ' for drag, ' L ' for lift, or ' m ' for pitching moment. The mean aerodynamic cord is given by \bar{c} and the aircraft speed relative to the air mass is given by V_T .

Similar to the longitudinal case, the lateral force and moment coefficients depend linearly on non-dimensionalized forms of the time rate of change of sideslip angle, the time rate of change of roll rate and the time rate of change of yaw rate. Let $\dot{\beta}$ denote the time rate of change of sideslip angle, \dot{p} denote the time rate of change of roll rate, and \dot{r} denote the time rate of change of yaw rate. The lateral force and moment coefficients, in stability frame, are given by

$$C_\chi = \dots + C_{\chi_{\dot{\beta}}} \left(\frac{\dot{\beta} b}{2V_T} \right) + C_{\chi_{\dot{p}}} \dot{p} \left(\frac{b}{2V_T} \right)^2 + C_{\chi_{\dot{r}}} \dot{r} \left(\frac{b}{2V_T} \right)^2, \quad (2.17)$$

where, for the lateral case, ' χ ' is replaced by ' Y ' for sideforce, ' l ' for rolling moment, or ' n ' for yawing moment. The aircraft wing span is denoted by b . The steady and unsteady flow aerodynamic forces and moments are transformed from the stability frame to the body frame before being added to the generalized force vector. This transformation process is described in Chapter 3.

To begin separating the aerodynamic forces and moments, first consider the angle of attack in still air, which is defined as

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right). \quad (2.18)$$

The time derivative of equation (2.18) is

$$\dot{\alpha} = \frac{u\dot{w} - w\dot{u}}{u^2 + w^2}. \quad (2.19)$$

The aircraft sideslip angle in still air is given by

$$\beta = \sin^{-1} \left(\frac{v}{V_T} \right). \quad (2.20)$$

The time derivative of equation (2.20) is

$$\dot{\beta} = \frac{V_T \dot{v} - v \dot{V}_T}{V_T \sqrt{u^2 + w^2}}. \quad (2.21)$$

Substituting equations (2.19) and (2.21) into equations (2.16) and (2.17) respectively, the aerodynamic forces and moments are clearly seen to be linear functions of body frame accelerations. Now, the generalized aerodynamic force vector $\{F\}_{Aero}$ can be expanded using Taylor Series as follows:

$$\{F\}_{Aero} = \{F\}_{Aero}|_{\{\dot{V}\}=0} + \frac{\partial \{F\}_{Aero}}{\partial \{\dot{V}\}} \bigg|_{\{\dot{V}\}=0} \{\delta \dot{V}\} + \frac{1}{2!} \frac{\partial^2 \{F\}_{Aero}}{\partial \{\dot{V}\}^2} \bigg|_{\{\dot{V}\}=0} \{\delta \dot{V}\}^2 + \dots \quad (2.22)$$

The linear dependence of the aerodynamic forces and moments on body frame accelerations simplifies equation (2.22) to

$$\{F\}_{Aero} = \{F\}_{Aero}|_{\{\dot{V}\}=0} + \frac{\partial \{F\}_{Aero}}{\partial \{\dot{V}\}} \bigg|_{\{\dot{V}\}=0} \{\dot{V}\}. \quad (2.23)$$

The generalized aerodynamic force vector is now separated into steady and unsteady components. Equation (2.12) can now be written as

$$[M]_I \{\dot{V}\} = -[\Omega][M]_I \{V\} + \{F\}_{Prop} + \{F\}_{Grav} + \{F\}_{Aero}|_{\{\dot{V}\}=0} - [M]_A \{\dot{V}\}, \quad (2.24)$$

where

$$[M]_A = - \frac{\partial \{F\}_{Aero}}{\partial \{\dot{V}\}} \bigg|_{\{\dot{V}\}=0} \quad (2.25)$$

$[M]_A$ represents an increment to the inertial mass matrix $[M]_I$, and $\{F\}_{Grav}$ consists of the gravity force vector defined in equation (2.5) and zero moment contributions to make the generalized gravity vector. Defining the generalized inertial force vector as

$$\{F\}_I = -[\Omega][M]_I \{V\}, \quad (2.26)$$

equation (2.24) can be written as

$$([M]_A + [M]_I) \{\dot{V}\} = \{F\}_I + \{F\}_{Prop} + \{F\}_{Grav} + \{F\}_{Aero}|_{\{\dot{V}\}=0}. \quad (2.27)$$

Finally, let the total mass matrix $[M]$ be given by

$$[M] = [M]_A + [M]_I. \quad (2.28)$$

The equations of motion as implemented in the simulation are given by

$$[M] \{\dot{V}\} = \{F\}_I + \{F\}_{Prop} + \{F\}_{Grav} + \{F\}_{Aero}|_{\{\dot{V}\}=0}. \quad (2.29)$$

At each integration time step equation (2.29) is solved for the unknown body frame accelerations by left-multiplying each side of the equation by the inverse of the total mass matrix.

2.6 Aircraft Mass Center Position

Let the aircraft mass center position relative to a round and rotating Earth be given by $[\tau \lambda h]'$ where τ defines longitude, λ defines latitude, and h defines altitude above sea level. Also, let the aircraft inertial translational velocity in local frame components be given by $[V_N V_E V_D]'$. The local horizontal reference frame velocity components are given by

$$\begin{Bmatrix} V_N \\ V_E \\ V_D \end{Bmatrix} = [T_{BV}]^{-1} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}. \quad (2.30)$$

The time rate of change of the aircraft position vector is given by

$$\begin{Bmatrix} \dot{\tau} \\ \dot{\lambda} \\ \dot{h} \end{Bmatrix} = \begin{Bmatrix} V_{Ec}/R \cos \lambda \\ V_N/R \\ -V_D \end{Bmatrix}, \quad (2.31)$$

where R is the distance from the center of the earth to the aircraft mass center and V_{Ec} is the eastward velocity component corrected for Earth rotation, which is given by

$$V_{Ec} = V_E - \omega_E R \cos \lambda. \quad (2.32)$$

A singularity occurs in equation (2.31) when λ is equal to 90° . The calculation of $\cos \lambda$ in equation (2.31) is therefore implemented in the simulation as

$$\cos \lambda = \begin{cases} \varepsilon \frac{\lambda}{|\lambda|} & |\cos \lambda| < \varepsilon \\ \cos \lambda & |\cos \lambda| \geq \varepsilon \end{cases}, \text{ where } \varepsilon = 10^{-10}. \quad (2.33)$$

Division by zero in equation (2.31) is therefore avoided. Equation (2.31) is integrated at each simulation time step to yield the aircraft position vector.

2.7 Aircraft Orientation Using Euler Angles

The orientation of the aircraft body reference frame with respect to a vehicle-carried local horizontal reference frame is described by the Euler angles ϕ , θ , and ψ . Let the relative rotational velocity between the body reference frame and the local horizontal reference frame be given by $[P Q R]'$. The time derivatives of the Euler angles are then calculated as

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{Bmatrix} P + \frac{1}{\cos \theta} (Q \sin \phi \sin \theta + R \cos \phi \sin \theta) \\ Q \cos \phi - R \sin \phi \\ \frac{1}{\cos \theta} (Q \sin \phi + R \cos \phi) \end{Bmatrix}, \quad (2.34)$$

where

$$\begin{Bmatrix} P \\ Q \\ R \end{Bmatrix} = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} - [T_{BV}] \begin{Bmatrix} (\omega_E + \dot{\tau}) \cos \lambda \\ -\dot{\lambda} \\ -(\omega_E + \dot{\tau}) \sin \lambda \end{Bmatrix}. \quad (2.35)$$

A singularity occurs in equation (2.34) when θ is equal to 90° . The calculation of $\cos \theta$ in equation (2.34) is implemented in the simulation as

$$\cos \theta = \begin{cases} \varepsilon \frac{\theta}{|\theta|} & |\cos \theta| < \varepsilon \\ \cos \theta & |\cos \theta| \geq \varepsilon \end{cases}, \text{ where } \varepsilon = 10^{-10}. \quad (2.36)$$

Division by zero in equation (2.34) is therefore avoided. Equation (2.30) is integrated at each simulation time step to yield the Euler angles.

2.8 Flight Path Calculations

Horizontal flight path angle, γ_h , and vertical flight path angle, γ_v , are given by

$$\gamma_h = \tan^{-1} \left(\frac{V_{Ec}}{V_N} \right) \quad (2.37)$$

$$\gamma_v = \tan^{-1} \left(\frac{-V_D}{\sqrt{V_{Ec}^2 + V_N^2}} \right). \quad (2.38)$$

2.9 Summary

Let the simulation state vector be defined as $\mathbf{x} = [u, v, w, p, q, r, h, \lambda, \tau, \phi, \theta, \psi]^T$. The equations required to propagate the state vector from t to $t + \Delta t$ are summarized in this section. The translational and rotational equations of motion of the aircraft are given by equation (2.29). The total mass matrix in equation (2.29), $[M]$, has two components, the inertial mass matrix $[M]_I$ and the apparent mass matrix $[M]_A$. The inertial mass matrix $[M]_I$ is defined by equation (2.13) and the computation of the apparent mass matrix $[M]_A$ is described in section 3.5. The inertial generalized force vector, $\{F\}_I$, is defined in equation (2.26) and the gravity force vector by equation (2.5). The calculation of the generalized aerodynamic force vector, $\{F\}_{Aero}|_{\{\dot{v}\}=0}$, is described in section 3.4. The computation of generalized propulsion force vector, $\{F\}_{Prop}$, is described in Chapter 4. The time rate of change of the aircraft's center of mass position is given by equation (2.31). The state vector and the transformations given by equations (2.30), (2.32), and (2.2) define the necessary quantities to propagate the aircraft's position. The time rate of change of the aircraft's orientation with respect to the local horizontal reference frame is given by equation (2.34). The state vector, equations (2.35), (2.31), and (2.2) define the necessary quantities to propagate the aircraft's angular position.

3 Aerodynamic Model

3.1 Aerodynamic Model Description

The Boeing Reference H cycle 1 aerodynamic force and moment coefficient buildup is discussed in detail in Dawdy et al - 1994. The rigid airframe aerodynamic force and moment coefficients are constructed from a collection of intermediate terms which are calculated from interpolated lookups of tabular data. The aerodynamic model includes a quasi-static aeroelastic logical variable, RIGID. If RIGID is set to false then quasi-static aeroelastic adjustments for flexibility are included in the aerodynamic force and moment coefficient buildup.

The aerodynamic model implemented in the simulation is divided into two parts. The steady flow part performs the aerodynamic force and moment coefficient buildup excluding the effects due to the rotational acceleration components, the time rate of change of the angle of attack, and the time rate of change of the sideslip angle. The calculation of steady flow aerodynamics is described in section 3.4. The unsteady flow part of the aerodynamic model accounts for the effects rotational acceleration, the time rate of change of the angle of attack, and the time rate of change of the sideslip angle with the apparent mass matrix and is described in section 3.5.

All aerodynamic terms are resolved about the stability axis and are referenced to the aerodynamic reference center which is defined in Table 3.1.

Table 3.1 Aerodynamic Reference Center Coordinates

Fuselage Station (+ aft)	2106. in
Wing Buttock Line (+ right)	0. in
Water Line (+ up)	226. in

3.2 Control Surfaces

Figure 3.1 defines the control surfaces for the cycle 1 aerodynamic model. The aerodynamic effects of the spoilers; control surfaces DSP1, DSP2, DSP3, and DSP4 shown in Figure 3.1, are not included in the cycle 1 aerodynamic model. The control surface deflection sign convention is given in Table 3.2.

3.3 Aerodynamic Force and Moment Calculations

The equations of motion described in Chapter 2 are derived for forces and moments acting at the aircraft center of gravity. As was mentioned previously, the cycle 1 aerodynamic model calculates forces and moments acting at the aerodynamic reference center. Following the work done by Buttrill et al - 1992, the inputs to the aerodynamic model are first transferred from the center of gravity to the aerodynamic reference center. The aerodynamic model is called and the aerodynamic forces and moments are then transferred from the aerodynamic reference center back to the center of gravity. The process of transferring the aerodynamic model inputs from the center of gravity to the aerodynamic reference center is described below.

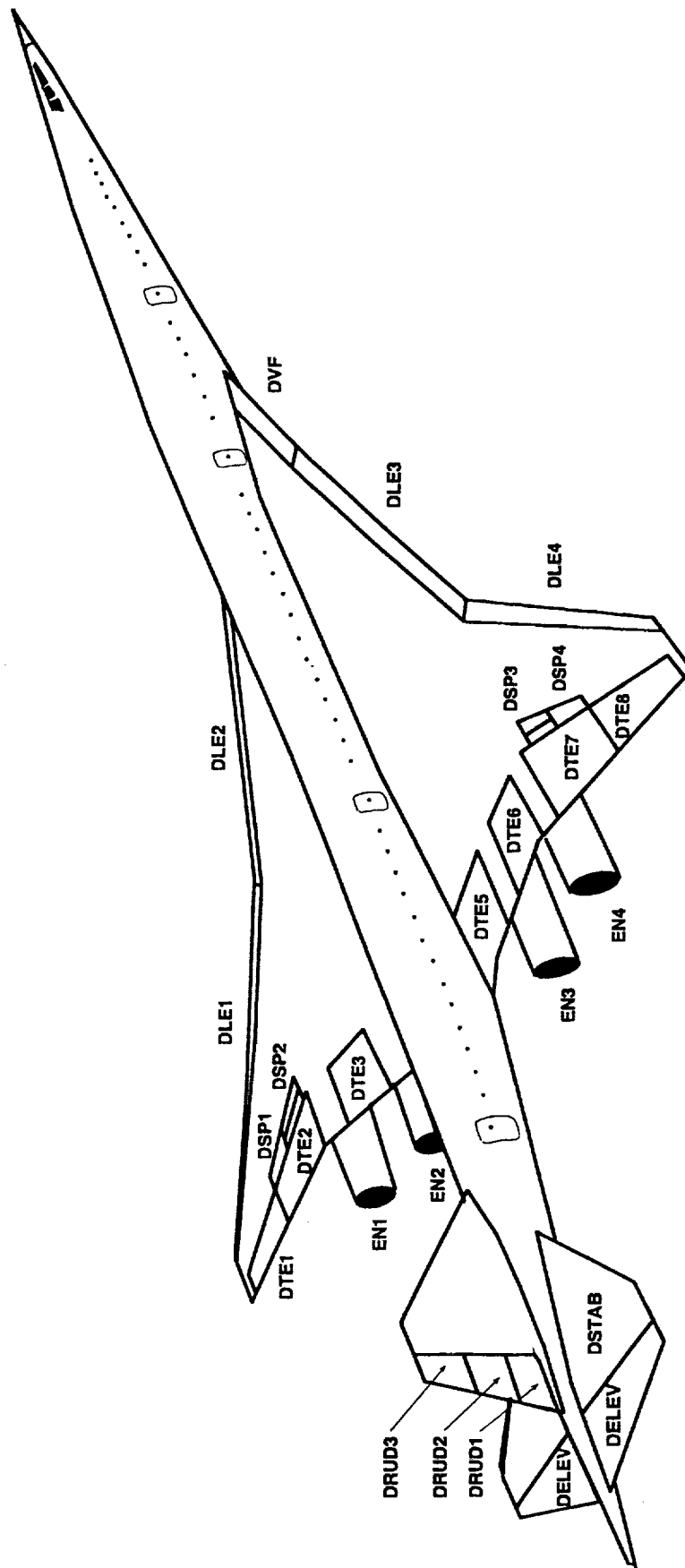


Figure 3.1 Reference H Configuration

Table 3.2 Control Surface Sign Convention

Control Surface	Description	Positive Deflection
DLE1	left outboard leading edge flap	leading edge down
DLE2	left inboard leading edge flap	leading edge down
DLE3	right inboard leading edge flap	leading edge down
DLE4	right outboard leading edge flap	leading edge down
DTE1	left outboard flaperon	trailing edge down
DTE2	left mid flaperon	trailing edge down
DTE3	left inboard flaperon	trailing edge down
DTE4	left root flap	trailing edge down
DTE5	right root flap	trailing edge down
DTE6	right inboard flaperon	trailing edge down
DTE7	right mid flaperon	trailing edge down
DTE8	right outboard flaperon	trailing edge down
DRUD1	lower rudder segment	trailing edge left
DRUD2	mid rudder segment	trailing edge left
DRUD3	upper rudder segment	trailing edge left
DSTAB	stabilizer	trailing edge down
DELEV	elevator	trailing edge down
DVF	vortex fence	trailing edge up

First the air relative translational velocity at the center of gravity is calculated by subtracting the velocity of the air mass due to wind, gusts, and turbulence from the inertial translational velocity of the aircraft at the center of gravity. The relative velocity components at the center of gravity are given by

$$\begin{Bmatrix} u_a \\ v_a \\ w_a \end{Bmatrix} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} - \begin{Bmatrix} u_{airmass} \\ v_{airmass} \\ w_{airmass} \end{Bmatrix}. \quad (3.1)$$

The relative translational velocity components are transferred to the aerodynamic reference center by

$$\begin{Bmatrix} u_{arf} \\ v_{arf} \\ w_{arf} \end{Bmatrix} = \begin{Bmatrix} u_a \\ v_a \\ w_a \end{Bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} x_{arf} \\ y_{arf} \\ z_{arf} \end{Bmatrix}, \quad (3.2)$$

where $[x_{arf} \ y_{arf} \ z_{arf}]'$ defines the distances from the aircraft center of gravity to the aerodynamic reference center along body frame axes. Positive x_{arf} locates the aerodynamic reference center ahead of the center of gravity, positive y_{arf} locates the aerodynamic reference center to the right of the center of gravity and positive z_{arf} locates the aerodynamic reference center below the center of gravity. The angle of attack at the aerodynamic center is computed as follows:

$$\alpha_{arf} = \tan^{-1} \left(\frac{w_{arf}}{u_{arf}} \right). \quad (3.3)$$

The sideslip angle at the aerodynamic reference center is given by

$$\beta_{arf} = \sin^{-1} \left(\frac{v_{arf}}{V_{T_{arf}}} \right), \quad (3.4)$$

where

$$V_{T_{arf}} = \sqrt{u_{arf}^2 + v_{arf}^2 + w_{arf}^2}. \quad (3.5)$$

The Mach number at the aerodynamic reference center is given by

$$M_{arf} = \frac{V_{T_{arf}}}{a}, \quad (3.6)$$

where a is the speed of sound. Dynamic pressure is left referenced to the center of gravity and is calculated as

$$\bar{q} = \frac{1}{2} \rho V_{T_a}^2, \quad (3.7)$$

where ρ is the air density in slugs/ft³ and V_{T_a} is relative airspeed at the center of gravity. The body frame rotational velocities at the aerodynamic reference center are assumed to be equal to the rotational velocities at the center of gravity.

3.4 Steady Flow Outputs

The quantities defined in equations (3.3) through (3.7) are combined with the altitude of the center of gravity, aircraft gross weight, body frame rotational velocities at the aerodynamic reference center, and control surface deflections to form the inputs to the aerodynamic model. The aerodynamic model defines the steady flow aerodynamic force and moment coefficients which are then used to calculate the following stability frame force and moment components:

$$\begin{Bmatrix} F_{X_{sf}} \\ F_{Y_{sf}} \\ F_{Z_{sf}} \\ M_{X_{sf}} \\ M_{Y_{sf}} \\ M_{Z_{sf}} \end{Bmatrix}_{arf}^{SA} = \bar{q} S \begin{Bmatrix} -C_{D_{sf}} \\ C_{Y_{sf}} \\ -C_{L_{sf}} \\ bC_{l_{sf}} \\ \bar{c}C_{m_{sf}} \\ bC_{n_{sf}} \end{Bmatrix}, \quad (3.8)$$

where S defines the aircraft wing reference area, the symbol 'SA' denotes stability frame, and the symbol 'sf' denotes steady flow. The stability frame components of the forces are then converted to body frame components as follows:

$$\begin{Bmatrix} F_{X_{sf}} \\ F_{Y_{sf}} \\ F_{Z_{sf}} \end{Bmatrix}_{arf}^{BA} = [T_{BS}] \begin{Bmatrix} F_{X_{sf}} \\ F_{Y_{sf}} \\ F_{Z_{sf}} \end{Bmatrix}_{arf}^{SA}, \quad (3.9)$$

where the symbol 'BA' denotes body frame and $[T_{BS}]$ is the transformation matrix from a stability reference frame to a body reference frame and is given by

$$[T_{BS}] = \begin{bmatrix} \cos \alpha_{arf} & 0 & -\sin \alpha_{arf} \\ 0 & 1 & 0 \\ \sin \alpha_{arf} & 0 & \cos \alpha_{arf} \end{bmatrix}. \quad (3.10)$$

Similarly, the body frame components of the steady flow aerodynamic moments are given by

$$\begin{Bmatrix} M_{X_{sf}} \\ M_{Y_{sf}} \\ M_{Z_{sf}} \end{Bmatrix}_{arf}^{BA} = [T_{BS}] \begin{Bmatrix} M_{X_{sf}} \\ M_{Y_{sf}} \\ M_{Z_{sf}} \end{Bmatrix}_{arf}^{SA}. \quad (3.11)$$

The body frame components of the steady flow aerodynamic forces at the center of gravity are equal to those at the aerodynamic reference center. The body frame components of the steady flow moments at the aircraft center of gravity are given as

$$\begin{Bmatrix} M_{X_{sf}} \\ M_{Y_{sf}} \\ M_{Z_{sf}} \end{Bmatrix}_{cg}^{BA} = \begin{Bmatrix} M_{X_{sf}} \\ M_{Y_{sf}} \\ M_{Z_{sf}} \end{Bmatrix}_{arf}^{BA} + \begin{bmatrix} 0 & -z_{arf} & y_{arf} \\ z_{arf} & 0 & -x_{arf} \\ -y_{arf} & x_{arf} & 0 \end{bmatrix} \begin{Bmatrix} F_{X_{sf}} \\ F_{Y_{sf}} \\ F_{Z_{sf}} \end{Bmatrix}_{arf}^{BA}. \quad (3.12)$$

Finally, the generalized steady flow aerodynamic force vector discussed in Section 2.5 is given by

$$\{F\}_{Aero} \big|_{\{\dot{v}\}=0} = \begin{Bmatrix} F_{X_{sf}} \\ F_{Y_{sf}} \\ F_{Z_{sf}} \\ M_{X_{sf}} \\ M_{Y_{sf}} \\ M_{Z_{sf}} \end{Bmatrix}_{cg}^{BA}. \quad (3.13)$$

3.5 Unsteady Flow Outputs

The aerodynamic effects of unsteady flow are included in the apparent mass matrix $[M]_A$. This section describes the calculations necessary to define the elements of the 6x6 apparent mass matrix. Taking the derivative with respect to time of equation (3.2) yields

$$\begin{Bmatrix} \dot{u}_{arf} \\ \dot{v}_{arf} \\ \dot{w}_{arf} \end{Bmatrix} = \begin{Bmatrix} \dot{u}_a - \dot{r}y_{arf} + \dot{q}z_{arf} \\ \dot{v}_a + \dot{r}x_{arf} - \dot{p}z_{arf} \\ \dot{w}_a - \dot{q}x_{arf} + \dot{p}y_{arf} \end{Bmatrix}. \quad (3.14)$$

The time rate of change of the angle of attack at the aerodynamic reference center is given by

$$\dot{\alpha}_{arf} = \frac{u_{arf}\dot{w}_{arf} - w_{arf}\dot{u}_{arf}}{u_{arf}^2 + w_{arf}^2}, \quad (3.15)$$

and the time rate of change of the sideslip angle at the aerodynamic reference center is given by

$$\dot{\beta}_{arf} = \frac{V_T\dot{v}_{arf} - v_{arf}\dot{V}_T}{V_T\sqrt{u^2 + w^2}}. \quad (3.16)$$

Substituting the expressions for \dot{u}_{arf} , \dot{v}_{arf} , and \dot{w}_{arf} given by equation (3.14) into equation (3.15) and equation (3.16) yields

$$\dot{\alpha}_{arf} = \frac{u_{arf}(\dot{w}_a - \dot{q}x_{arf} + \dot{p}y_{arf}) - w_{arf}(\dot{u}_a - \dot{r}y_{arf} + \dot{q}z_{arf})}{u_{arf}^2 + w_{arf}^2} \quad (3.17)$$

and

$$\dot{\beta}_{arf} = \frac{\dot{v}_a + \dot{r}x_{arf} - \dot{p}z_{arf}}{\sqrt{u_{arf}^2 + w_{arf}^2}} - \frac{v_{arf}(u_{arf}\dot{u}_{arf} + v_{arf}\dot{v}_{arf} + w_{arf}\dot{w}_{arf})}{V_T^2\sqrt{u_{arf}^2 + w_{arf}^2}}. \quad (3.18)$$

Transforming the $\dot{\alpha}$ and $\dot{\beta}$ stability derivatives from stability frame components to body frame components yields

$$C_{X_\eta} = C_{L_\eta} \sin \alpha_{arf} - C_{D_\eta} \cos \alpha_{arf} \quad (3.19)$$

$$C_{Z_\eta} = -C_{L_\eta} \cos \alpha_{arf} - C_{D_\eta} \sin \alpha_{arf}, \quad (3.20)$$

where ' η ' is replaced by ' $\dot{\alpha}$ ' or ' $\dot{\beta}$ '. The body frame components of the unsteady flow aerodynamic forces at the aerodynamic reference center are given below:

$$F_{X_{usf}} = \bar{q}SC_{X_{\dot{\alpha}_{arf}}} \left(\frac{\dot{\alpha}_{arf}\bar{c}}{2V_T} \right) + \bar{q}SC_{X_{\dot{\beta}_{arf}}} \dot{\beta}_{arf} \left(\frac{\bar{c}}{2V_T} \right)^2 \quad (3.21)$$

$$F_{Y_{usf}} = \bar{q}SC_{Y_{\dot{\beta}_{arf}}} \left(\frac{\dot{\beta}_{arf}b}{2V_T} \right) + \bar{q}SC_{Y_{\dot{p}}} \dot{p} \left(\frac{b}{2V_T} \right)^2 + \bar{q}SC_{Y_{\dot{r}}} \dot{r} \left(\frac{b}{2V_T} \right)^2 \quad (3.22)$$

$$F_{Z_{usf}} = \bar{q}SC_{Z_{\dot{\alpha}_{arf}}} \left(\frac{\dot{\alpha}_{arf}\bar{c}}{2V_T} \right) + \bar{q}SC_{Z_{\dot{q}}} \dot{q} \left(\frac{\bar{c}}{2V_T} \right)^2. \quad (3.23)$$

Similarly, transforming the $\dot{\beta}$, \dot{p} , and \dot{r} stability derivatives from stability frame components to body frame components gives

$$C_{\kappa}^{BA} = C_{\kappa}^{SA} \cos \alpha_{arf} - C_{n_{\kappa}}^{SA} \sin \alpha_{arf} \quad (3.24)$$

$$C_{n_{\kappa}}^{BA} = C_{\kappa}^{SA} \sin \alpha_{arf} + C_{n_{\kappa}}^{SA} \cos \alpha_{arf}, \quad (3.25)$$

where ' κ ' is replaced by ' $\dot{\beta}$ ', ' \dot{p} ', or ' \dot{r} '. The body frame components of the unsteady flow aerodynamic moments about the aerodynamic reference center are given below:

$$M_{X_{usf}} = \bar{q}SbC_{l_{\dot{\beta}_{arf}}} \left(\frac{\dot{\beta}_{arf}b}{2V_T} \right) + \bar{q}SbC_{l_{\dot{p}}} \dot{p} \left(\frac{b}{2V_T} \right)^2 + \bar{q}SbC_{l_{\dot{r}}} \dot{r} \left(\frac{b}{2V_T} \right)^2 \quad (3.26)$$

$$M_{Y_{usf}} = \bar{q}S\bar{c}C_{m_{\dot{\alpha}_{arf}}} \left(\frac{\dot{\alpha}_{arf}\bar{c}}{2V_T} \right) + \bar{q}S\bar{c}C_{m_{\dot{q}}} \dot{q} \left(\frac{\bar{c}}{2V_T} \right)^2 \quad (3.27)$$

$$M_{Z_{usf}} = \bar{q}SbC_{n_{\dot{\beta}_{arf}}} \left(\frac{\dot{\beta}_{arf}b}{2V_T} \right) + \bar{q}SbC_{n_{\dot{p}}} \dot{p} \left(\frac{b}{2V_T} \right)^2 + \bar{q}SbC_{n_{\dot{r}}} \dot{r} \left(\frac{b}{2V_T} \right)^2, \quad (3.28)$$

where the symbol ' usf ' denotes unsteady flow. The expressions for $\dot{\alpha}_{arf}$ and $\dot{\beta}_{arf}$ given in equation (3.17) and equation (3.18), respectively, can be substituted into the equations (3.21) through (3.23) and equations (3.26) through (3.28) to yield the unsteady flow force and moment components as functions of body frame accelerations. Transferring the unsteady flow forces from the aerodynamic reference center to the center of gravity gives

$$\begin{Bmatrix} F_{X_{usf}} \\ F_{Y_{usf}} \\ F_{Z_{usf}} \end{Bmatrix}_{cg}^{BA} = \begin{Bmatrix} F_{X_{usf}} \\ F_{Y_{usf}} \\ F_{Z_{usf}} \end{Bmatrix}_{arf}^{BA}. \quad (3.29)$$

The unsteady flow aerodynamic moments in body frame components at the center of gravity are given by

$$\begin{Bmatrix} M_{X_{usf}} \\ M_{Y_{usf}} \\ M_{Z_{usf}} \end{Bmatrix}_{cg}^{BA} = \begin{Bmatrix} M_{X_{usf}} \\ M_{Y_{usf}} \\ M_{Z_{usf}} \end{Bmatrix}_{arf}^{BA} + \begin{bmatrix} 0 & -z_{arf} & y_{arf} \\ z_{arf} & 0 & -x_{arf} \\ -y_{arf} & x_{arf} & 0 \end{bmatrix} \begin{Bmatrix} F_{X_{usf}} \\ F_{Y_{usf}} \\ F_{Z_{usf}} \end{Bmatrix}_{arf}^{BA}. \quad (3.30)$$

The apparent mass matrix, defined by equation (2.25), is represented as

$$[M]_A = - \begin{bmatrix} \frac{\partial F_{X_{usf}}}{\partial \dot{u}_a} & \frac{\partial F_{X_{usf}}}{\partial \dot{v}_a} & \frac{\partial F_{X_{usf}}}{\partial \dot{w}_a} & \frac{\partial F_{X_{usf}}}{\partial \dot{p}} & \frac{\partial F_{X_{usf}}}{\partial \dot{q}} & \frac{\partial F_{X_{usf}}}{\partial \dot{r}} \\ \frac{\partial F_{Y_{usf}}}{\partial \dot{u}_a} & \frac{\partial F_{Y_{usf}}}{\partial \dot{v}_a} & \frac{\partial F_{Y_{usf}}}{\partial \dot{w}_a} & \frac{\partial F_{Y_{usf}}}{\partial \dot{p}} & \frac{\partial F_{Y_{usf}}}{\partial \dot{q}} & \frac{\partial F_{Y_{usf}}}{\partial \dot{r}} \\ \frac{\partial F_{Z_{usf}}}{\partial \dot{u}_a} & \frac{\partial F_{Z_{usf}}}{\partial \dot{v}_a} & \frac{\partial F_{Z_{usf}}}{\partial \dot{w}_a} & \frac{\partial F_{Z_{usf}}}{\partial \dot{p}} & \frac{\partial F_{Z_{usf}}}{\partial \dot{q}} & \frac{\partial F_{Z_{usf}}}{\partial \dot{r}} \\ \frac{\partial M_{X_{usf}}}{\partial \dot{u}_a} & \frac{\partial M_{X_{usf}}}{\partial \dot{v}_a} & \frac{\partial M_{X_{usf}}}{\partial \dot{w}_a} & \frac{\partial M_{X_{usf}}}{\partial \dot{p}} & \frac{\partial M_{X_{usf}}}{\partial \dot{q}} & \frac{\partial M_{X_{usf}}}{\partial \dot{r}} \\ \frac{\partial M_{Y_{usf}}}{\partial \dot{u}_a} & \frac{\partial M_{Y_{usf}}}{\partial \dot{v}_a} & \frac{\partial M_{Y_{usf}}}{\partial \dot{w}_a} & \frac{\partial M_{Y_{usf}}}{\partial \dot{p}} & \frac{\partial M_{Y_{usf}}}{\partial \dot{q}} & \frac{\partial M_{Y_{usf}}}{\partial \dot{r}} \\ \frac{\partial M_{Z_{usf}}}{\partial \dot{u}_a} & \frac{\partial M_{Z_{usf}}}{\partial \dot{v}_a} & \frac{\partial M_{Z_{usf}}}{\partial \dot{w}_a} & \frac{\partial M_{Z_{usf}}}{\partial \dot{p}} & \frac{\partial M_{Z_{usf}}}{\partial \dot{q}} & \frac{\partial M_{Z_{usf}}}{\partial \dot{r}} \end{bmatrix}, \quad (3.31)$$

where the analytical expressions for the elements of the apparent mass matrix are defined in Appendix A.

3.6 Aerodynamic Model Implementation

The aerodynamic model is a FORTRAN S-function which is compiled as a MATLAB FMEX file. The aerodynamic model FMEX file is dynamically linked into the simulation during run time. The input parameters for the aerodynamic model are given in Table 3.3. INIT_FLAG is an initialization variable. If INIT_FLAG is set equal to 1, all the cycle 1 aerodynamic data is read into memory. If INIT_FLAG is 0, then the data is not read into memory. The variable RIGID_FLAG is used to set the value of the logical variable RIGID. Setting RIGID_FLAG equal to 1 assigns RIGID to true. If RIGID_FLAG is set equal to 0, RIGID is false. Table 3.4 lists the aerodynamic model outputs.

Table 3.3 Aerodynamic Model Input Parameters

Variable	Description	Symbol
INIT_FLAG	initialization flag	
RIGID_FLAG	quasi-static aeroelastic flag	
U_ARF	x-body component of inertial translational velocity at arc (ft/s)	u_{arf}
V_ARF	y-body component of inertial translational velocity at arc (ft/s)	v_{arf}
W_ARF	z-body component of inertial translational velocity at arc (ft/s)	w_{arf}
PBA_AR	x-body component of rotational velocity at arc (ft/s)	p
QBA_AR	y-body component of rotational velocity at arc (ft/s)	q
RBA_AR	z-body component of rotational velocity at arc (ft/s)	r
X_ARF	distance from cg to arc along x-body axis (ft)	x_{arf}
Y_ARF	distance from cg to arc along y-body axis (ft)	y_{arf}
Z_ARF	distance from cg to arc along z-body axis (ft)	z_{arf}
ALPDEG	angle-of-attack at arc (degrees)	α_{arf}
BETADEG	sideslip angle at arc (degrees)	β_{arf}
VTOTAL	airspeed at the arc (ft/s)	V_T
MACH	Mach number	M_{arf}
QBAR	dynamic pressure at cg (lbf/ft ²)	\bar{q}
ALT	altitude of cg (ft)	h
DLE1	left outboard leading edge flap deflection (degrees)	DLE1
DLE2	left inboard leading edge flap deflection (degrees)	DLE2
DLE3	right inboard leading edge flap deflection (degrees)	DLE3
DLE4	right outboard leading edge flap deflection (degrees)	DLE4
DTE1	left outboard flaperon deflection (degrees)	DTE1
DTE2	left mid flaperon deflection (degrees)	DTE2
DTE3	left inboard flaperon deflection (degrees)	DTE3
DTE4	left root flap deflection (degrees)	DTE4
DTE5	right root flap deflection (degrees)	DTE5
DTE6	right inboard flaperon deflection (degrees)	DTE6
DTE7	right mid flaperon deflection (degrees)	DTE7
DTE8	right outboard flaperon deflection (degrees)	DTE8
DRUD1	lower rudder segment deflection (degrees)	DRUD1
DRUD2	mid rudder segment deflection (degrees)	DRUD2
DRUD3	upper rudder segment deflection (degrees)	DRUD3
DSTAB	stabilizer deflection (degrees)	DSTAB
DELEV	elevator deflection (degrees)	DELEV
DVF	vortex fence deflection (degrees)	DVF
DGEAR	landing gear extension (degrees)	DGEAR
GW	gross weight (lbf)	

Table 3.4 Aerodynamic Model Output Parameters

Variable	Definition	Symbol
MA	apparent mass matrix (see Appendix A)	$[M]_A$
FXBA_FM	x-body component of steady flow aerodynamic forces (lbf)	$F_{X_{sf}}$
FYBA_FM	y-body component of steady flow aerodynamic forces (lbf)	$F_{Y_{sf}}$
FZBA_FM	z-body component of steady flow aerodynamic forces (ft-lbf)	$F_{Z_{sf}}$
LMBA_FM	x-body component of steady flow aerodynamic moments (ft-lbf)	$M_{X_{sf}}$
MMBA_FM	y-body component of steady flow aerodynamic moments (ft-lbf)	$M_{Y_{sf}}$
NMBA_FM	z-body component of steady flow aerodynamic moments (ft-lbf)	$M_{Z_{sf}}$
PSA_AR	x-stability component of rotational velocity at arc (ft/s)	
QSA_AR	y-stability component of rotational velocity at arc (ft/s)	
RSA_AR	z-stability component of rotational velocity at arc (ft/s)	
CD	steady flow drag coefficient at arc	$C_{D_{sf}}$
CL	steady flow lift coefficient at arc	$C_{L_{sf}}$
CM	steady flow pitching moment coefficient at arc	$C_{m_{sf}}$
CR	steady flow rolling moment coefficient at arc	$C_{l_{sf}}$
CY	steady flow sideforce coefficient at arc	$C_{Y_{sf}}$
CN	steady flow yawing moment coefficient at arc	$C_{n_{sf}}$

4 Engine Model

4.1 Engine Model Description

The engine model used in the simulation is representative of a supersonic transport engine. The engine is a mixed flow turbofan which incorporates a downstream mixer, 50 % aspiration ratio, a nozzle, and a mixed-compression translating centerbody inlet. All four engines are shown in Figure 3.1. The model data covers altitudes from sea level to 70,000 feet and Mach numbers from 0 to 2.4. Gross thrust and ram drag are not included in the model; rather, installed net thrust is used. Inlet forces are not specifically modeled, therefore this model does not realistically simulate engine failures. The engine model is discussed in Dawdy et al - 1994.

4.2 Engine Model Dynamics

The engine model simulates second order nonlinear engine dynamics. The engine dynamics are shown in Figure 4.1. The cockpit power lever angle range is limited to values between 0° and 100°. The limited power lever angle is then directed into a lead-lag filter followed by a rate limiter and a lag filter. The filtered cockpit power lever angle value from each throttle is then used to obtain the net installed thrust for each engine.

4.3 Engine Model Buildup Equations

The engine model calculates the steady state or instantaneous installed net thrust produced by each engine for a given power setting. Maximum and minimum installed net thrust limits are calculated from interpolated tabular data which is a function of altitude and Mach number. The thrust limits are then used with filtered cockpit power lever angle to compute installed net thrust per engine. The total installed net thrust from all four engines is used to compute the total thrust-induced body frame forces and moments about the aerodynamic reference center. The aerodynamic reference center location is defined in Table 3.1.

4.4 Engine Model Implementation

The engine model is implemented in two parts. The transfer function implementation of the engine dynamics is shown in Figure 4.2. Cockpit power lever angle for each throttle is fed into the engine dynamics. The power lever angle signals are biased to zero using the initial cockpit power lever angle values. The bias is used to initially set the signal to zero since the filter states are initially set to zero in SIMULINK. The biased cockpit power lever angle signals are limited accordingly and then directed into the lead-lag filter, followed by the rate limiter, and the lag filter. The bias is then removed from the signals, yielding the filtered cockpit power lever angles which are then input to the second part of the engine model. The second part of the engine model calculates thrust-induced body frame forces and moments about the aerodynamic reference center. This part of the engine model is a FORTRAN S-function which is compiled as a MATLAB FMEX file. The FMEX file is dynamically linked to the simulation during run time. The engine model input parameters are given in Table 4.1. The variable TLIMIT_FLAG is used to set the logical variable TLIMIT. If TLIMIT_FLAG is set equal to 1, TLIMIT is set to true. Setting TLIMIT to true will

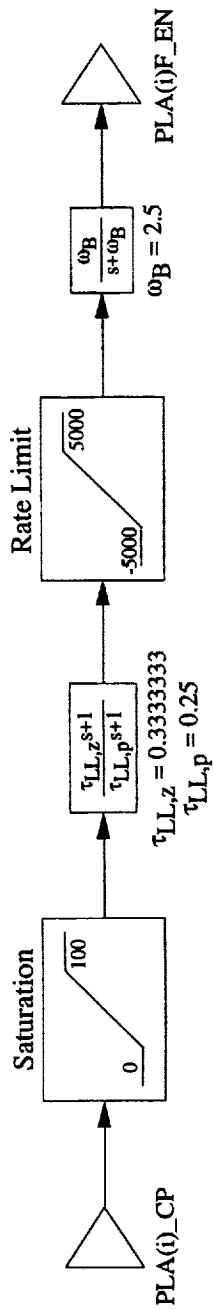


Figure 4.1 Reference H Cycle 1 Engine Dynamics

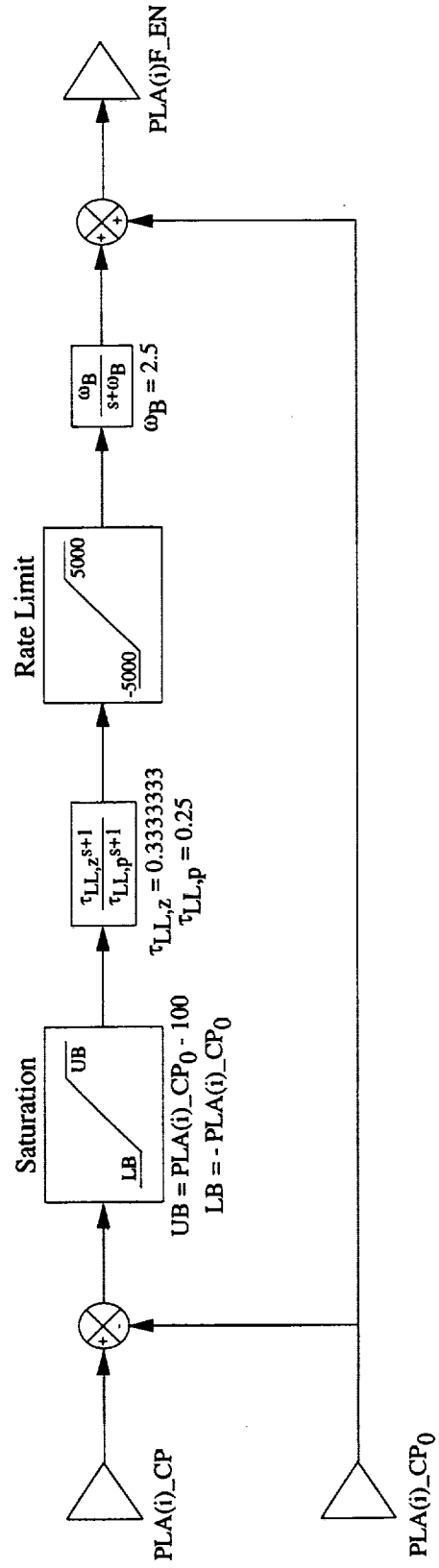


Figure 4.2 Reference H Cycle 1 Engine Dynamics SIMULINK Implementation

cause the engine model to calculate maximum and minimum installed net thrust as a function of altitude and Mach number. If TLIMIT_FLAG is set equal to 0, then TLIMIT is set to false and maximum installed net thrust is set equal to 100,000 lbf and minimum installed net thrust is set to 0 lbf. Table 4.2 gives the engine model output parameters.

Table 4.1 Engine Model Input Parameters

	Variable	Description	Symbol
Part 1	PLA1_CP	throttle 1 cockpit power lever angle (deg)	PLA1_CP
	PLA2_CP	throttle 2 cockpit power lever angle (deg)	PLA2_CP
	PLA3_CP	throttle 3 cockpit power lever angle (deg)	PLA3_CP
	PLA4_CP	throttle 4 cockpit power lever angle (deg)	PLA4_CP
Part 2	TLIMIT_FLAG	maximum and minimum installed net thrust flag	
	HPRESS_AT	altitude (ft)	h
	MACH_AR	Mach number	M
	PLA1F_EN	throttle 1 filtered cockpit power lever angle (deg)	PLA1F_EN
	PLA2F_EN	throttle 2 filtered cockpit power lever angle (deg)	PLA2F_EN
	PLA3F_EN	throttle 3 filtered cockpit power lever angle (deg)	PLA3F_EN
	PLA4F_EN	throttle 4 filtered cockpit power lever angle (deg)	PLA4F_EN

Table 4.2 Engine Model Output Parameters

	Variable	Description	Symbol
Part 1	PLA1F_EN	throttle 1 filtered cockpit power lever angle (deg)	PLA1F_EN
	PLA2F_EN	throttle 2 filtered cockpit power lever angle (deg)	PLA2F_EN
	PLA3F_EN	throttle 3 filtered cockpit power lever angle (deg)	PLA3F_EN
	PLA4F_EN	throttle 4 filtered cockpit power lever angle (deg)	PLA4F_EN
Part 2	FXBT_FM	x-body component of thrust-induced forces (lbf)	$F_{X_{prop}}$
	FYBT_FM	y-body component of thrust-induced forces (lbf)	$F_{Y_{prop}}$
	FZBT_FM	z-body component of thrust-induced forces (lbf)	$F_{Z_{prop}}$
	MLBT_FM	x-body component of thrust-induced moments (ft-lbf)	$M_{X_{prop}}$
	MMBT_FM	y-body component of thrust-induced moments (ft-lbf)	$M_{Y_{prop}}$
	MNBT_FM	z-body component of thrust-induced moments (ft-lbf)	$M_{Z_{prop}}$
	FN1_EN	installed net thrust for engine 1 (lbf)	
	FN2_EN	installed net thrust for engine 2 (lbf)	
	FN3_EN	installed net thrust for engine 3 (lbf)	
	FN4_EN	installed net thrust for engine 4 (lbf)	

5 Mass Model

5.1 Mass Model Description

The Boeing Reference H cycle 1 simulation mass model is discussed in detail in Dawdy et al - 1994. Mass dynamics are not modeled in the simulation since the mass model does not include fuel consumption data. The model contains center of gravity and moment of inertia data for seven mass sets. These mass sets are listed in Table 5.1:

Table 5.1 Mass Model Mass Distributions

Mass Set	Description	Gross Weight
M01	operating empty weight	279,080 (lbf)
M05	M01 + full wing fuel	646,458 (lbf)
M13	maximum taxi weight at forward cg	649,914 (lbf)
M14	MZFW + aft body tank full + payload for aft cg	402,466 (lbf)
MIC	initial cruise condition	614,864 (lbf)
MCR	mid cruise design point	501,324 (lbf)
MFC	final cruise condition	384,862 (lbf)

MZFW denotes maximum zero fuel weight.

5.2 Mass Model Implementation

The mass model consists of FORTRAN code, which is compiled as a MATLAB FMEX file and dynamically linked to the simulation at run time as an S-function. The input parameters to the mass model are given in Table 5.2:

Table 5.2 Mass Model Input Parameters

Variable	Description	Symbol
AUTO CG_FLAG	cg location flag	
AUTO WT_FLAG	mass set flag	
GW	gross weight (lbf)	
GRAV_AT	gravitational acceleration (ft/s ²)	g
XCG_PMAC	longitudinal cg location in percent mean aerodynamic chord	
YCG_BL_MD	lateral cg wing buttock line location (in)	
ZCG_WL_MD	vertical cg waterline location (in)	

The input variable AUTOWT_FLAG is used to set the logical variable AUTOWT. If AUTOWT_FLAG is equal to 1, AUTOWT is set to true and the mass model buildup equations will use data associated with one of the seven mass sets. Gross weight will be set to the gross weight for the associated mass set. A zero value for AUTOWT_FLAG causes AUTOWT to be set to false. The gross weight input value is used in the table lookups to allow interpolation between data points. The input variable AUTO CG_FLAG is used to set the logical variable AUTO CG. If AUTO CG_FLAG is set equal to 1, AUTO CG is set to true and the mass model uses the center of gravity position associated with the appropriate mass set. An AUTO CG_FLAG value of 0 sets AUTO CG to false. Setting AUTO CG to false will cause the mass model to accept the longitudinal center of gravity location as a percent of mean aerodynamic chord, lateral center of gravity wing

buttock line location, and vertical center of gravity waterline location as additional inputs. The model uses these values to calculate the center of gravity position; however, it may provide inaccurate moment of inertia parameters for a given mass set. The mass model outputs are listed in Table 5.3:

Table 5.3 Mass Model Output Parameters

Variable	Definition	Symbol
DXCG_MD	distance from cg to arc along x-body axis (ft)	x_{arf}
DYCG_MD	distance from cg to arc along y-body axis (ft)	y_{arf}
DZCG_MD	distance from cg to arc along z-body axis (ft)	z_{arf}
WT_MD	gross weight (lbf)	
MASS_MD	mass (slug)	m
IXX_MD	roll moment of inertia (slug-ft ²)	I_x
IYY_MD	pitch moment of inertia (slug-ft ²)	I_y
IZZ_MD	yaw moment of inertia (slug-ft ²)	I_z
IXZ_MD	roll-yaw product of inertia (slug-ft ²)	I_{xz}
XCG_PMAC	longitudinal cg location in percent mean aerodynamic chord	
XCG_BS_MD	longitudinal cg body station location (in)	
YCG_BL_MD	lateral cg wing buttock line location (in)	
ZCG_WL_MD	vertical cg waterline location (in)	

6 Flight Control System Model

6.1 Accelerometers

The simulation performs calculations for axial, lateral, and vertical accelerometers. Accelerometer output is in g units. Axial accelerometer output is positive for a forward acceleration, lateral accelerometer output is positive for an acceleration to the right, and vertical accelerometer output is positive for an upward acceleration. The output for accelerometers positioned at the aircraft center of gravity is given by

$$\begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} = \frac{1}{g} \begin{Bmatrix} \dot{u} + qw - vr + g \sin \theta \\ \dot{v} + ru - pw - g \cos \theta \sin \phi \\ -(\dot{w} + pv - qu - g \cos \theta \cos \phi) \end{Bmatrix}. \quad (6.1)$$

For accelerometers positioned at any other location, the accelerations with lever arm effects included and the 1 g bias removed are given by

$$\begin{Bmatrix} n_{x_s} \\ n_{y_s} \\ n_{z_s} \end{Bmatrix} = \frac{1}{g} \begin{bmatrix} -(q^2 + r^2) & pq - \dot{r} & pr + \dot{q} \\ pq + \dot{r} & -(p^2 + r^2) & qr - \dot{p} \\ pr - \dot{q} & qr + \dot{p} & -(p^2 + q^2) \end{bmatrix} \begin{Bmatrix} x_s \\ y_s \\ z_s \end{Bmatrix} + \begin{Bmatrix} n_x \\ n_y \\ -1 + n_z \end{Bmatrix}, \quad (6.2)$$

where sensor location relative to the center of gravity in the body reference frame, in feet, is given by $[x_s, y_s, z_s]^T$. Let x_s represent distance forward from the center of gravity, y_s represent distance to the right of the center of gravity, and z_s represent distance below the center of gravity. Therefore, sensor location is given by

$$\begin{Bmatrix} x_s \\ y_s \\ z_s \end{Bmatrix} = \frac{1}{12} \begin{Bmatrix} -(BS_s - BS_{cg}) \\ (BL_s - BL_{cg}) \\ -(WL_s - WL_{cg}) \end{Bmatrix}, \quad (6.3)$$

where BS , BL , and WL denote locations, in inches, along body station, wing buttock line, and waterline respectively.

6.2 Actuators

The Reference H control surfaces are hydraulically powered. Actuator dynamics are not modeled in the simulation. Inclusion of an actuator model will be part of future model development.

6.3 Flight Control System

The simulation models the open-loop, bare airframe and does not include a flight control system. Addition of a flight control system model will be part of future model development.

7 Atmosphere Model

7.1 1962 Standard Atmosphere Model

The simulation uses the 1962 Standard Atmosphere Tables for density and speed of sound. The tables contain data in cubic spline format for altitudes ranging from sea level to 240,000 feet. The interpolated values for speed of sound, density, and density ratio are calculated as a function of altitude. The atmosphere model is written in C and is compiled as a CMEX file. During run time, the CMEX file is dynamically linked to the simulation. The atmosphere model is taken from LaRCsim (Jackson - 1995).

7.2 Dryden Turbulence Model

A simple atmospheric turbulence model, based on the Dryden spectra (Hoblit - 1988), is made available in the simulation for the purpose of providing an atmospheric disturbance input to the aircraft's dynamics. Dryden atmospheric turbulence can be simulated in the continuous time domain by passing the band-limited white noise signal through the following filters:

$$\text{Longitudinal: } F_u(s) = \sqrt{\frac{\pi}{\Delta t}} \sqrt{\frac{2V\sigma_u^2}{L_u\pi}} \frac{1}{\frac{V}{L_u} + s} \quad (7.1)$$

$$\text{Lateral and Vertical: } F_w(s) = \sqrt{\frac{\pi}{\Delta t}} \sqrt{\frac{3V\sigma_w^2}{L_w\pi}} \frac{\left(\frac{V}{L_w\sqrt{3}} + s\right)}{\left(\frac{V}{L_w} + s\right)^2}. \quad (7.2)$$

Here, V is the vehicle's total air-relative velocity, $\sigma_{u,w}^2$ are the variance of longitudinal and lateral-vertical turbulence, respectively, $L_{u,w}$ are the spatial scale lengths for longitudinal and lateral-vertical turbulence, respectively, and Δt is the simulation integration time step. The values of $\sigma_{u,w}^2$ and $L_{u,w}$ used in the simulation are a function of altitude and are, for the case of severe turbulence (Johnson et al - 1993), given in Table 7.1. Note that the metric units given in Table 7.1 are converted to English units in the implementation of these filters in the simulation. To generate the three components of turbulence, three distinct uncorrelated Gaussian white noise sources are used.

Table 7.1 RMS values and Scale Lengths for longitudinal and lateral-vertical severe turbulence as a function of altitude.

Altitude (MSL) km	RMS values		Scale Lengths	
	Longitudinal σ_u , m/s	Lateral and Vertical σ_w , m/s	Longitudinal L_u , km	Lateral and Vertical L_w , km
1	5.70	4.67	0.832	0.624
2	5.80	4.75	0.902	0.831
4	6.24	5.13	1.04	0.972
6	7.16	5.69	1.04	1.01
8	7.59	5.98	1.04	0.98
10	7.72	6.00	1.23	1.10
12	7.89	5.71	1.80	1.54
14	6.93	5.05	2.82	2.12
16	5.00	4.31	3.40	2.60
18	4.07	3.81	5.00	3.34
20	3.85	3.38	8.64	4.41
25	4.34	3.34	12.0	6.56
30	5.60	3.59	28.6	8.88

8 Simulation Structure

8.1 Simulation Layout

Figure 8.1 shows a top level layout of the nonlinear simulation. The simulation is initialized at time $t = 0$ seconds by the initial state vector \mathbf{x}_0 . At time $t > 0$ seconds, the state vector which drives the simulation is switched from the initial state vector to the state vector calculated during the previous time step. As seen in Figure 8.1, all calculations are performed in nine separate subsystems. These subsystems are discussed below.

Atmospheric disturbances are modeled in the *Wind and Turbulence* subsystem. This subsystem maybe used to specify steady wind velocity components in a local horizontal reference frame and atmospheric turbulence velocity components in body axis system, which are used to determine the air mass velocity at the aircraft center of gravity. A Dryden atmospheric turbulence model is provided in this sub-block which maybe replaced by other atmospheric turbulence models. The airmass velocity components are then passed into the *Auxiliary Equations* subsystem which calculates the relative velocity at the aircraft center of gravity. The relative velocity components are transferred to the aerodynamic reference center as discussed in Section 3.3. Auxillary quantities needed by the simulation and the aircraft model are computed in this subsystem. The 1962 Standard Atmosphere model is used in this subsystem.

The aircraft is modeled by the *Mass*, *Control Inputs*, *Propulsion*, and *Aerodynamics* subsystems. The *Mass* subsystem executes the Reference H mass model and also calculates the body frame gravitational forces acting on the aircraft. The *Control Inputs* subsystem sets the control surface positions needed for the evaluation of the Reference H aerodynamic model and the throttle settings for the Reference H engine model. The Reference H engine model is called in the *Propulsion* subsystem. Finally, the aircraft model is completed by the *Aerodynamics* subsystem which executes the Reference H aerodynamic model.

The *External* subsystem calculates the total external and inertial forces and moments acting on the aircraft at the center of gravity. The propulsive and steady flow aerodynamic forces and moments are transferred from the aerodynamic reference center to the center of gravity and added to the gravitational force and inertial force components (equation (2.26)) to yield the right-hand-side of equation (2.29). The *Equations of Motion* subsystem solves equation (2.29) for the acceleration vector. The six remaining state derivatives are calculated as discussed in Sections 2.6 and 2.7. The state derivative vector is then integrated to give the current state vector, which is fed back for use in the next simulation time step.

The *Acceleration* subsystem calculates accelerometer output at the center of gravity and at a specified sensor location.

8.2 Simulation Implementation

The SIMULINK implementation of the nonlinear simulation is shown in Figure 8.2. The simulation contains ten subsystems. Each subsystem is discussed below.

The *Initialization* subsystem utilizes a switch to control which state vector is used in the simulation. At time $t = 0$ seconds, the initial simulation states are set by the vector **InitialStates**, which is

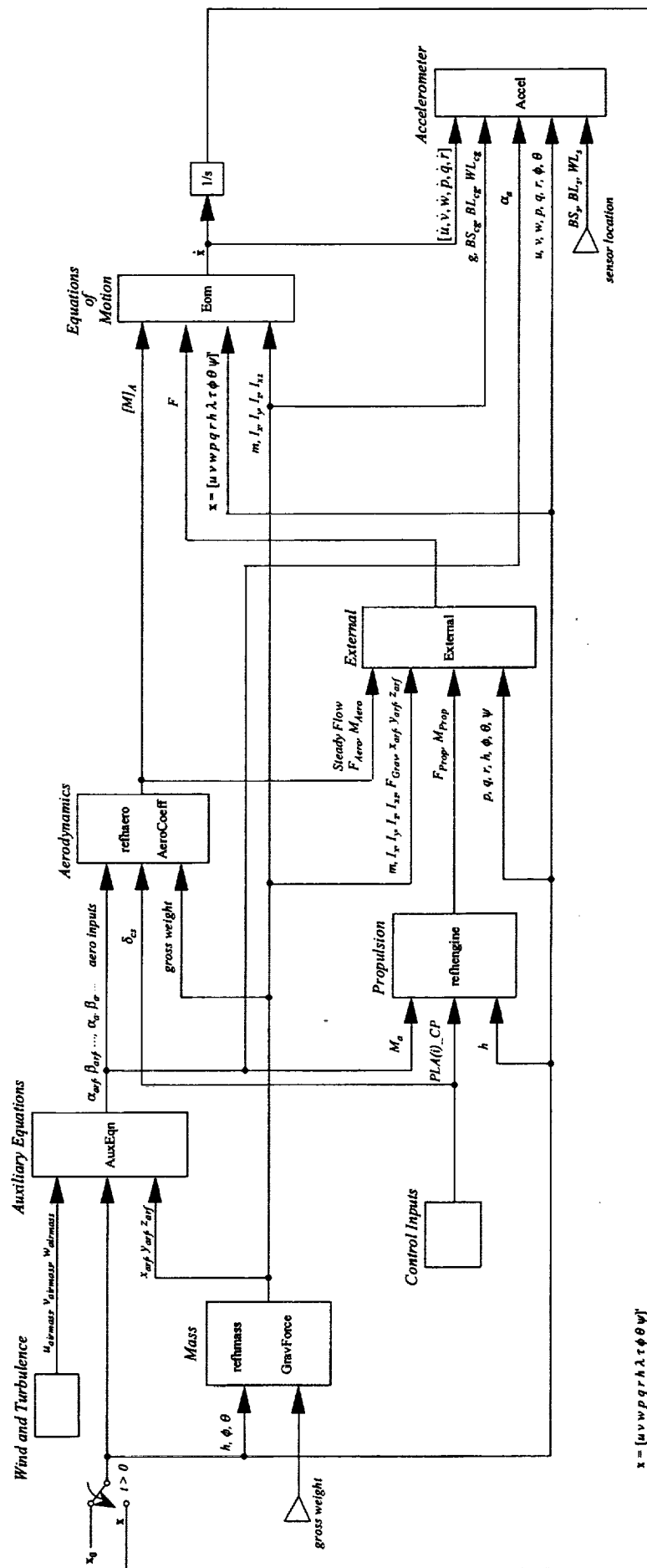


Figure 8.1 Nonlinear Simulation Layout

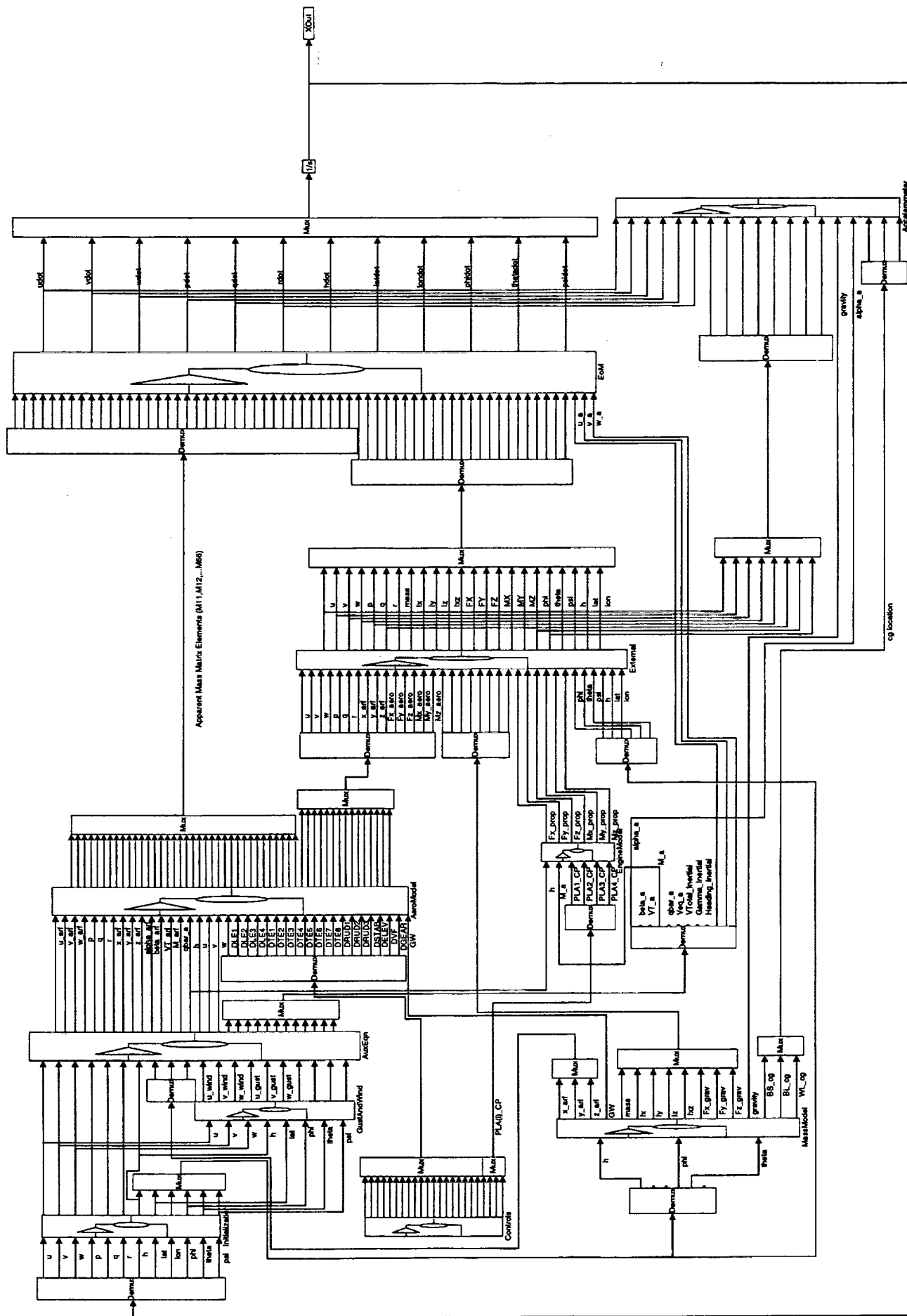


Figure 8.2 Nonlinear Simulation SIMULINK Implementation – refhsim.m

defined in the MATLAB workspace. At time $t \geq 0.000001$ seconds, the state vector from the previous time step is used. Simulation time is also maintained in this subsystem. The *GustAndWind* subsystem uses the Dryden turbulence model to simulate atmospheric disturbances. This subsystem calls the functions *DrydenVel* and *dryden_model_coeff*. Steady wind velocities in local horizontal coordinate system can be set in this subsystem and atmospheric turbulence velocity components in body axes corresponding to severe turbulence levels are calculated. The variable **ROTATION_FLAG**, which is defined in the MATLAB workspace, is used to include Earth's rotational effects in the equations of motion. If **ROTATION_FLAG** is set to 1, Earth rotation is included in the computations. If **ROTATION_FLAG** is set to 0, the computations do not include Earth rotation. The variable **DRYDEN_FLAG** is also defined in the MATLAB workspace. If **DRYDEN_FLAG** is set to zero, the Dryden turbulence model is bypassed and gust velocities are set to zero. If **DRYDEN_FLAG** is set equal to 1, then the Dryden turbulence model is used to compute translational gust velocities.

The *AuxEqn* subsystem calls the function *AuxEqn*, which computes all necessary quantities at the center of gravity and at the aerodynamic reference center. *AuxEqn* also executes the 1962 Standard Atmosphere model function, *atmosphere_62*.

The *Controls* subsystem sets the control positions that are necessary for the Reference H aerodynamic and engine models. The control inputs are set by the 22x1 vector **control**, which is defined in the MATLAB workspace.

The *MassModel* subsystem executes the Reference H cycle 1 mass model, *refhmass*, and also calls the function *GravForce*. The function *GravForce* calculates the body frame components of the gravitational force. Gross weight is set by the variable **GW**, which is defined in the MATLAB workspace. The mass model logical variables are set by the variables **AUTOWT_FLAG** and **AUTO CG_FLAG**, which are also defined in the MATLAB workspace. If **AUTO CG_FLAG** is 0, center of gravity position is set by defining the variables **XCG_PMAC**, **YCG_BL_MD**, and **ZCG_WL_MD** in the workspace.

Engine dynamics are implemented in the *EngineModel* subsystem. This subsystem also executes the engine model *refhengine*. The engine model logical variable **TLIMIT** is set by the variable **TLIMIT_FLAG**, which is defined in the MATLAB workspace.

The *AeroModel* subsystem executes the Reference H cycle 1 aerodynamic model *refhaero*. The aerodynamic model logical variables are set by the variables **INIT_FLAG** and **RIGID_FLAG**, which are defined in the MATLAB workspace. This subsystem calls the function *AeroCoeff*, which computes the steady flow aerodynamic force and moment coefficients at the center of gravity.

The *External* subsystem calls the function *External*, which computes the total external and inertial forces and moments acting at the center of gravity. The variable **ROTATION_FLAG** is used in this subsystem.

The *EoM* subsystem calls the function *Eom*, which executes the equations of motion. This function calculates the state derivative vector as well as flight path angle, horizontal flight path angle, time rate of change of angle of attack at the center of gravity, time rate of change of sideslip angle, and time rate of change of total airspeed at the center of gravity. The variable **ROTATION_FLAG** is also used in this subsystem.

The last subsystem, *Accelerometer*, calls the function *Accel* to compute accelerometer output. Sensor location is set by the 3x1 vector **sensor**, which is defined in the MATLAB workspace.

8.3 Simulation Initialization And Setup

A MATLAB m-file can be used to initialize the aerodynamic model for the nonlinear simulation. An example m-file for initialization is given in Figure 8.3.

```
% INITIALIZATION

UAeroTemp = [
1.;           % INIT_FLAG (1 -> yes, 0 -> no)
0;           % RIGID_FLAG (0 -> elastic, 1 -> rigid)
250.;        % U_ARF (ft/s)
50.;         % V_ARF (ft/s)
-100.;       % W_ARF (ft/s)
0.1;         % PSA_AR (rad/sec)
-0.1;        % QSA_AR (rad/sec)
0.1;         % RSA_AR (rad/sec)
50.;         % DXCG_MD (ft)
1.;          % DYCG_MD (ft)
-2.;         % DZCG_MD (ft)
12.;         % ALPHAA_AR (deg)
-6.;         % BETAA_AR (deg)
800.;        % VTOTAL (ft/s)
0.4;         % MACH
300.;        % QBAR_AR (psf)
50.;         % HR_ER (ft)
10.;         % DLE1 (deg)
10.;         % DLE2 (deg)
20.;         % DLE3 (deg)
10.;         % DLE4 (deg)
10.;         % DTE1 (deg)
30.;         % DTE2 (deg)
-10.;        % DTE3 (deg)
30.;         % DTE4 (deg)
30.;         % DTE5 (deg)
20.;         % DTE6 (deg)
30.;         % DTE7 (deg)
-20.;        % DTE8 (deg)
5.;          % DRUD1 (deg)
-5.;         % DRUD2 (deg)
5.;          % DRUD3 (deg)
5.;          % DSTAB (deg)
-15.;        % DELEV (deg)
40.;         % DVF (deg)
60.;         % DGEAR (deg)
501324.;     % GW or WT_MD (lbs)
];

YAeroTemp = refhaero(0, [], UAeroTemp, 3);

clear UAeroTemp YAeroTemp;
```

Figure 8.3 Example Initialization M-file - aero_init.m

UAeroTemp is an arbitrary input vector used to initialize the aerodynamic model. The only element that has any significance is the first element, which is used to set the initialization variable. This element must be set equal to 1. The command line

$$\text{YAeroTemp} = \text{refhaero}(0, [], \text{UAeroTemp}, 3); \quad (8.1)$$

initializes the aerodynamic model with UAeroTemp by loading all aerodynamic data into memory. The initialization process takes approximately 120 seconds. The flight condition of interest is set by defining the variables **GW**, **INIT_FLAG**, **RIGID_FLAG**, **AUTO CG_FLAG**, **AUTOWT_FLAG**, **TLIMIT_FLAG**, **ROTATION_FLAG**, **DRYDEN_FLAG**, **XCG_PMAC**, **YCG_PMAC**, and **ZCG_WL_MD** in the MATLAB workspace. The variable **INIT_FLAG** should be set to 0, otherwise the aerodynamic model will be initialized in each time step. The vectors **InitialStates**, **control**, and **sensor** must also be defined in the MATLAB workspace. *Please note that if any of the workspace variables are changed during a session, the SIMULINK block diagram must be reinitialized.* This reinitialization is accomplished by typing the following command line at the MATLAB prompt:

```
sizes = refhsim; (8.2)
```

An example m-file for setting the simulation is given in Figure 8.4:

```
% SET FLIGHT CONDITION FOR YAW DOUBLET DYNAMIC CHECK CASE
```

```
GW = 279080.;
INIT_FLAG = 0.;
RIGID_FLAG = 0.;
AUTOWT_FLAG = 1.;
AUTO CG_FLAG = 1.;
TLIMIT_FLAG = 1.;
ROTATION_FLAG = 0.;
DRYDEN_FLAG = 0.;
XCG_PMAC = 54.602;
YCG_BL_MD = 0.;
ZCG_WL_MD = 201.5;

InitialStates = [
1036.57435;           % u (ft/s)
0.;                  % v (ft/s)
27.092043;           % w (ft/s)
0.;                  % p (rad/s)
0.;                  % q (rad/s)
0.;                  % r (rad/s)
20000.;              % h (ft)
0.*pi/180.;          % lat (rad)
0.*pi/180.;          % lon (rad)
-7.58562275e-22*pi/180.; % phi (rad)
1.497179975*pi/180.; % theta (rad)
0.000319998*pi/180.; % psi (rad)
];

control = [
10.;                % DLE1 (deg)
0.;                 % DLE2 (deg)
0.;                 % DLE3 (deg)
10.;                % DLE4 (deg)
3.;                 % DTE1 (deg)
3.;                 % DTE2 (deg)
0.;                 % DTE3 (deg)
0.;                 % DTE4 (deg)
0.;                 % DTE5 (deg)
0.;                 % DTE6 (deg)
3.;                 % DTE7 (deg)
3.;                 % DTE8 (deg)
0.;                 % DRUD1 (deg)
0.;                 % DRUD2 (deg)
0.;                 % DRUD3 (deg)
-.426949;           % DSTAB (deg)
];
```



```

-.853898;          % DELEV (deg)
0.;                % DVF (deg)
0.;                % DGEAR (deg)
68.4916            % PLA1_CP (deg)
68.4916            % PLA2_CP (deg)
68.4916            % PLA3_CP (deg)
68.4916            % PLA4_CP (deg)
];

% Sensor Location
sensor = [
150.;              % BS sensor (in)
-40.;              % BL sensor (in)
-36;               % WL sensor (in)
];

```

Figure 8.4 Example Flight Condition Setup M-file - yawdoub_setup.m

8.4 Simulation Results

Time history results are given in Appendix B for the flight condition defined in Figure 8.4, with a 10 degree doublet applied to all three rudder segments. Table 8.1 lists the constants which are hard coded into the simulation. In this example simulation, it is assumed that there are no atmospheric disturbances and that Earth's rotation is zero. The simulation was run at 40 Hz using an Euler integration algorithm. The simulation took 122.444 seconds to run. These time history results are compared with those obtained from independent simulations at the Boeing Commercial Aircraft Company and the real-time VMS (Visual Motion Simulator) simulation at NASA Langley Research Center.

Table 8.1 Constants used in the simulation

Symbol	Description	Numerical Value
b	wing span reference length, ft	129.643
\bar{c}	wing chord reference length, ft	86.023
g_0	gravitational acceleration at sea level, ft/s ²	32.228
R_E	Earth radius, ft	20,898,908
S	wing reference area, ft ²	7100.0
ω^E	Earth rotation, rad/s	7.2722e-05

9 Trim Solutions

9.1 Finding A Trim Solution

In order to describe the process for finding a trim solution, consider a simple longitudinal level-flight trim problem. For the nonlinear simulation, this trim problem can be stated as follows:

Find

$$\mathbf{X} = [\delta_{stab} \quad \delta_e \quad \delta_{pla1} \quad \delta_{pla2} \quad \delta_{pla3} \quad \delta_{pla4} \quad u \quad w \quad \theta] \quad (9.1)$$

such that

$$\mathbf{G} = [M - M_{des} \quad \gamma_v - \gamma_{vdes} \quad \dot{V}_T \quad \dot{\alpha} \quad \dot{q}] = \{0\} \quad (9.2)$$

for a given flight condition. Stabilizer deflection and elevator deflection are denoted by δ_{stab} and δ_e respectively. The four throttle positions are defined by δ_{pla1} , δ_{pla2} , δ_{pla3} , and δ_{pla4} . These throttle positions are equivalent to the cockpit power lever angles discussed in Chapter 4. The subscript 'des' denotes desired value. Independent parameters in equation (9.1) provide the necessary degrees of freedom required to satisfy the nonlinear constraints specified in equation (9.2), i.e., to perform a longitudinal trim of the aircraft at the desired flight condition.

9.2 Trim Solution Implementation

If the elevator is considered to be geared to the stabilizer, then if the stabilizer deflection required to trim is known, the corresponding elevator deflection is also known. Therefore, elevator deflection required to trim can be removed from in equation (9.1). Also, all four throttles are assumed to have equal settings and therefore only one throttle position is required in \mathbf{X} . With these simplifications, the longitudinal level-flight trim problem can now be stated as follows:

Find

$$\mathbf{X} = [\delta_{stab} \quad \delta_{pla1} \quad u \quad w \quad \theta] \quad (9.3)$$

such that

$$\mathbf{G} = [M - M_{des} \quad \gamma_v - \gamma_{vdes} \quad \dot{V}_T \quad \dot{\alpha} \quad \dot{q}] = \{0\} \quad (9.4)$$

for a given flight condition. The SIMULINK block diagram used for finding a trim solution is shown in Figure 9.1. Engine dynamics and atmospheric disturbances are not modeled. Figure 9.2 gives an example m-file used to define and solve the longitudinal level-flight trim problem:



```

% SETUP AND SOLVE LONGITUDINAL LEVEL-FLIGHT TRIM PROBLEM

% SET WORKSPACE VARIABLES
GW = 501324;
INIT_FLAG = 0.;
RIGID_FLAG = 0.;
AUTO CG_FLAG = 1;
AUTOWT_FLAG = 1;
TLIMIT_FLAG = 1.;
ROTATION_FLAG = 0.;
XCG_PMAC = 48.0;
YCG_BL_MD = 0.0;
ZCG_WL_MD = 226.0;
ZCG_WL_MD = 190.0;

% SET SENSOR LOCATION
sensor = [
150.;           % BS (in)
-30.;           % BL (in)
-36.;           % WL (in)
];

% REINITIALIZE BLOCK DIAGRAM
sizes = refhsim_trim;

% SET CONTROL INPUTS (minus DSTAB, DELEV, and PLA(i)_CP)
u = [
10.;           % 1 - DLE1 (deg)
0.;           % 2 - DLE2 (deg)
0.;           % 3 - DLE3 (deg)
10.;          % 4 - DLE4 (deg)
3.;           % 5 - DTE1 (deg)
3.;           % 6 - DTE2 (deg)
0.;           % 7 - DTE3 (deg)
0.;           % 8 - DTE4 (deg)
0.;           % 9 - DTE5 (deg)
0.;           % 10 - DTE6 (deg)
3.;           % 11 - DTE7 (deg)
3.;           % 12 - DTE8 (deg)
0.;           % 13 - DRUD1 (deg)
0.;           % 14 - DRUD2 (deg)
0.;           % 15 - DRUD3 (deg)
0.;           % 16 - DVF (deg)
0.;           % 17 - DGEAR (deg)
];

% SET STATES (minus u, w, and theta)
x = [
0.;           % 1 - v (ft/s)
0.;           % 2 - p (rad/s)
0.;           % 3 - q (rad/s)
0.;           % 4 - r (rad/s)
30000.;       % 5 - h (ft)
0.*pi/180.;   % 6 - lat (rad)
0.*pi/180.;   % 7 - lon (rad)
0.*pi/180.;   % 8 - phi (rad)
-140.*pi/180.; % 9 - psi (rad)
];

% SET DESIRED CONSTRAINT VALUES
g_des = [
0.8;          % Mach
0.0*pi/180.;  % gamma
0.0;          % VT_dot

```

```

0.0*pi/180;          % alpha_dot
0.0*pi/180;          % q_dot
];

% SET OPTIONS VECTOR FOR CONSTR FUNCTION
%           1  2  3  4  5 6 7 8 9 10 11 12 13 14 15 16 17 18
options = [ 1 1e-4 1e-4 1e-6 0 0 0 0 0 0 0 0 0 5 1000 0 1e-6 1.0 1.0]';

% SET INITIAL GUESS, LOWER BOUNDS, AND UPPER BOUNDS for X
%           DSTAB  PLA_CP  u      w      Theta
Xstart = [ 3.      50      900    65.    2.0*pi/180]';
Xlb     = [ -15     0       0     -500   -15*pi/180 ]';
Xub     = [ 15     100    2600    500    35*pi/180 ]';

% FIND TRIM SOLUTION
Xfinal = constr('longtrim_fun',Xstart,options,Xlb,Xub,'',u,x,g_des)

% COMPUTE OBJECTIVE FUNCTION, CONSTRAINT VECTOR, AND OUTPUT VECTOR
% AT TRIMMED CONDITIONS
[f,G,Y]=longtrim_fun(Xfinal,u,x,g_des)

% DISPLAY RESULTS
fprintf('Longitudinal, Level-Flight Trim Solution\n\n');
fprintf('DSTAB = %10.5f      DELEV = %10.5f      PLA_CP=%10.5f\n', ...
        Xfinal(1),2*Xfinal(2));
fprintf('u = %10.5f      w = %10.5f      theta = %10.5f\n', ...
        Xfinal(3),Xfinal(4),Xfinal(5)*180/pi);
fprintf('M = %10.5f      gamma = %10.5f      VT_dot = %10.5f\n', ...
        Y(3),Y(7)*180/pi,Y(23));
fprintf('alpha_dot = %10.5f      qdot = %10.5f\n',Y(21),Y(15));

```

Figure 9.2 Example M-File For Computing A Longitudinal Trim Solution - longtrim.m

Please see the *Optimization Toolbox User's Guide* for a detailed explanation of the vector **options**. The command line

sizes = refhsim_trim; (9.5)

reinitializes the SIMULINK block diagram as discussed in Section 8.2. A constrained nonlinear optimization is performed by the MATLAB function *constr* on the function *longtrim_fun*. The function file *longtrim_fun.m* is given in Figure 9.3. The outputs of the trim process are the elements of the trim input vector **X**, the block diagram output vector **Y**, the constraint vector **G**, and the objective function **f**.

```

function [f,G,Y] = longtrim_fun(X,u,x,g_des)
% FIND TRIM SOLUTION

% SET INITIAL CONDITIONS
x0 = [];

u0 = [
u(1);          % 1 - DLE1 (deg)
u(2);          % 2 - DLE2 (deg)
u(3);          % 3 - DLE3 (deg)
u(4);          % 4 - DLE4 (deg)
u(5);          % 5 - DTE1 (deg)
u(6);          % 6 - DTE2 (deg)
u(7);          % 7 - DTE3 (deg)
u(8);          % 8 - DTE4 (deg)
u(9);          % 9 - DTE5 (deg)
u(10);         % 10 - DTE6 (deg)

```

```

u(11);          % 11 - DTE7 (deg)
u(12);          % 12 - DTE8 (deg)
u(13);          % 13 - DRUD1 (deg)
u(14);          % 14 - DRUD2 (deg)
u(15);          % 15 - DRUD3 (deg)
X(1);           % 16 - DSTAB (deg)
2*X(1);         % 17 - DELEV (deg)  ->  DELEV = 2*DSTAB
u(16);          % 18 - DVF (deg)
u(17);          % 19 - DGEAR (deg)
X(2);           % 20 - PLA1_CP (deg)
%              - PLA2_CP (deg)
%              - PLA3_CP (deg)
%              - PLA4_CP (deg)
X(3);           % 21 - u (ft/s)
x(1);           % 22 - v (ft/s)
X(4);           % 23 - w (ft/s)
x(2);           % 24 - p (rad/s)
x(3);           % 25 - q (rad/s)
x(4);           % 26 - r (rad/s)
x(5);           % 27 - h (ft)
x(6);           % 28 - lat (rad)
x(7);           % 29 - lon (rad)
x(8);           % 30 - phi (rad)
X(5);           % 31 - theta (rad)
x(9);           % 32 - psi (rad)
];

% CALCULATE OUTPUTS
Y = refhsim_trim(0,x0,u0,3);

% CALCULATE CONSTRAINTS AND OBJECTIVE FUNCTION
G(1) = Y(3) - g_des(1);      % Mach Number constraint
G(2) = Y(7) - g_des(2);      % Flight Path angle constraint
G(3) = Y(23) - g_des(3);     % VT_dot constraint
G(4) = Y(21) - g_des(4);     % alpha_dot constraint
G(5) = Y(15) - g_des(5);     % q_dot constraint
f = 0.0;

```

Figure 9.3 Longitudinal Trim Function *longtrim_fun* - longtrim_fun.m

10 Linear Models

10.1 Introduction

This chapter describes the methodology by which linear models are generated from the nonlinear simulation. Linear model of the aircraft dynamics can be obtained from the nonlinear simulation in MATLAB/SIMULINK environment by using the MATLAB's *linmod* command. However, since the nonlinear simulation uses the body axes equations of motion, the linear models generated are body axes linear models, which are typically not convenient for controls engineering. Therefore, the body axes state-space linear models are transformed to a set of preferred states and outputs. This chapter describes that transformation process and presents an example case.

10.2 States, Inputs, and Outputs

Let the nonlinear simulation state vector, input vector, and output vector be given by $\bar{\mathbf{x}}$, $\bar{\mathbf{u}}$, and $\bar{\mathbf{y}}$ respectively. Also, let \mathbf{x} define a new state vector with the preferred states, \mathbf{u} redefines the input vector, and \mathbf{y} define a new output vector with the preferred outputs. The state vectors, input vectors, and output vectors previously mentioned are defined in Table 10.1. For the engine dynamics, τ_{LL} represents the lead-lag filter state and ω_B represents the lag filter state. γ_h denotes the horizontal flight path angle, γ_v denotes the vertical flight path angle, n_y denotes the lateral acceleration at the center of gravity, n_z denotes the vertical acceleration at the center of gravity, n_{y_s} denotes the lateral acceleration at the pilot station, and n_{z_s} denotes the vertical acceleration at the pilot station.

10.3 State Vector and Output Vector Transformations

Looking at the two state vectors, the states V_T , α , and β replace the states u , v , and w . Recall that total airspeed, angle of attack, and sideslip angle are functions of the body frame components of inertial translational velocity; therefore, the two state vectors are related as follows:

$$\mathbf{x} = \mathbf{T}_x \bar{\mathbf{x}}, \quad (10.1)$$

where \mathbf{T}_x is some undefined transformation matrix. Similarly, the output vectors are related by

$$\mathbf{y} = \mathbf{T}_y \bar{\mathbf{y}}, \quad (10.2)$$

where \mathbf{T}_y is another undefined transformation matrix. To define \mathbf{T}_x and \mathbf{T}_y , recall that total airspeed is given by

$$V_T = \sqrt{u^2 + v^2 + w^2}. \quad (10.3)$$

Expanding V_T about V_{T_0} using Taylor series gives

$$\Delta V_T = V_{T_u} \Delta u + \frac{1}{2!} V_{T_{uu}} (\Delta u)^2 + \dots + V_{T_v} \Delta v + \frac{1}{2!} V_{T_{vv}} (\Delta v)^2 + \dots + V_{T_w} \Delta w + \frac{1}{2!} V_{T_{ww}} (\Delta w)^2 + \dots \quad (10.4)$$

Table 10.1 State, Input, and Output Vectors

Element	$\tilde{\mathbf{x}}$	$\tilde{\mathbf{u}}$	$\tilde{\mathbf{y}}$	\mathbf{x}	\mathbf{u}	\mathbf{y}
1	ω_{B4}	DLE1	u	V_T	DLE1	V_T
2	τ_{LL4}	DLE2	v	α	DLE2	α
3	ω_{B3}	DLE3	w	q	DLE3	q
4	τ_{LL3}	DLE4	p	θ	DLE4	θ
5	ω_{B2}	DTE1	q	h	DTE1	h
6	τ_{LL2}	DTE2	r	p	DTE2	p
7	ω_{B1}	DTE3	h	r	DTE3	r
8	τ_{LL1}	DTE4	ϕ	ϕ	DTE4	ϕ
9	u	DTE5	θ	β	DTE5	β
10	v	DTE6	ψ	ψ	DTE6	ψ
11	w	DTE7	τ_{LL1}	λ	DTE7	τ_{LL1}
12	p	DTE8	ω_{B1}	τ	DTE8	ω_{B1}
13	q	DRUD1	τ_{LL2}	τ_{LL1}	DRUD1	τ_{LL2}
14	r	DRUD2	ω_{B2}	ω_{B1}	DRUD2	ω_{B2}
15	h	DRUD3	τ_{LL3}	τ_{LL2}	DRUD3	τ_{LL3}
16	λ	DSTAB	ω_{B3}	ω_{B2}	DSTAB	ω_{B3}
17	τ	DELEV	τ_{LL4}	τ_{LL3}	DELEV	τ_{LL4}
18	ϕ	DVF	ω_{B4}	ω_{B3}	DVF	ω_{B4}
19	θ	DGEAR	γ_h	τ_{LL4}	DGEAR	γ_h
20	ψ	PLA1_CP	γ_v	ω_{B4}	PLA1_CP	γ_v
21		PLA2_CP	n_y		PLA2_CP	n_y
22		PLA3_CP	n_z		PLA3_CP	n_z
23		PLA4_CP	n_{y_s}		PLA4_CP	n_{y_s}
24			n_{z_s}			n_{z_s}
25			\dot{h}			\dot{h}
26			$F_{X_{sf}}$			$F_{X_{sf}}$
27			$F_{Y_{sf}}$			$F_{Y_{sf}}$
28			$F_{Z_{sf}}$			$F_{Z_{sf}}$
29			$M_{X_{sf}}$			$M_{X_{sf}}$
30			$M_{Y_{sf}}$			$M_{Y_{sf}}$
31			$M_{Z_{sf}}$			$M_{Z_{sf}}$

Keeping only the linear terms from equation (10.4) yields

$$\Delta V_T = V_{T_u} \Delta u + V_{T_v} \Delta v + V_{T_w} \Delta w, \quad (10.5)$$

where

$$V_{T_u} = \left. \frac{\partial V_T}{\partial u} \right|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}_0} \quad (10.6,a)$$

$$V_{T_v} = \left. \frac{\partial V_T}{\partial v} \right|_{\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_0} \quad (10.6,b)$$

$$V_{T_w} = \left. \frac{\partial V_T}{\partial w} \right|_{\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_0} \quad (10.6,c)$$

Equation (10.5) relates the linear variation in total airspeed to the linear variations in inertial translational velocity body frame components. If the procedure just described is applied to angle of attack and sideslip angle, the linear variation in angle of attack about α_0 is given as

$$\Delta\alpha = \alpha_u \Delta u + \alpha_w \Delta w \quad (10.7)$$

and the linear variation in sideslip angle about β_0 is given by

$$\Delta\beta = \beta_u \Delta u + \beta_v \Delta v + \beta_w \Delta w. \quad (10.8)$$

Now the transformation matrices can be defined. The transformation matrix \mathbf{T}_x is given by

$$\mathbf{T}_x = \left[\begin{array}{c|c} \mathbf{0}_{12 \times 8} & \mathbf{T}_{s_{12 \times 12}} \\ \hline \mathbf{T}_{e_{8 \times 8}} & \mathbf{0}_{8 \times 12} \end{array} \right], \quad (10.9)$$

where

$$\mathbf{T}_s = \begin{bmatrix} V_{T_u} & V_{T_v} & V_{T_w} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha_u & 0 & \alpha_w & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \beta_u & \beta_v & \beta_w & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (10.10)$$

and

$$\mathbf{T_e} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (10.11)$$

The elements of $\mathbf{T_s}$ which are functions of inertial body frame acceleration are given by

$$V_{T_u} = \frac{u}{V_T} \quad (10.12,a)$$

$$V_{T_v} = \frac{v}{V_T} \quad (10.12,b)$$

$$V_{T_w} = \frac{w}{V_T} \quad (10.12,c)$$

$$\alpha_u = \frac{-w}{u^2 \left(1 + \left(\frac{w}{u} \right)^2 \right)} \quad (10.12,d)$$

$$\alpha_w = \frac{1}{u \left(1 + \left(\frac{w}{u} \right)^2 \right)} \quad (10.12,e)$$

$$\beta_u = \frac{-uv}{V_T^3 \sqrt{1 - \left(\frac{v}{V_T} \right)^2}} \quad (10.12,f)$$

$$\beta_v = \frac{-1}{\sqrt{1 - \left(\frac{v}{V_T} \right)^2}} \left(\frac{1}{V_T} - \frac{v^2}{V_T^3} \right) \quad (10.12,g)$$

$$\beta_w = \frac{-vw}{V_T^3 \sqrt{1 - \left(\frac{v}{V_T} \right)^2}}. \quad (10.12,h)$$

The transformation matrix \mathbf{T}_y is given by

$$\mathbf{T}_y = \left[\begin{array}{c|c} \mathbf{T}_{0_{10 \times 10}} & \mathbf{0}_{10 \times 21} \\ \hline \mathbf{0}_{21 \times 10} & \mathbf{I}_{21 \times 21} \end{array} \right], \quad (10.13)$$

where

$$\mathbf{T}_0 = \begin{bmatrix} V_{T_u} & V_{T_v} & V_{T_w} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_u & 0 & a_w & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \beta_u & \beta_v & \beta_w & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10.14)$$

10.4 State-Space Form

Let the linear model generated from the nonlinear simulation be given in state-space form by

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}\tilde{\mathbf{u}} \quad (10.15,a)$$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{C}}\tilde{\mathbf{x}} + \tilde{\mathbf{D}}\tilde{\mathbf{u}}. \quad (10.15,b)$$

Let the transformed linear model with the desired states and outputs be given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (10.16,a)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \quad (10.16,b)$$

This section describes how equation (10.15) is transformed to give equation (10.16).

Leftmultiplying both sides of equation (10.1) by \mathbf{T}_x^{-1} gives

$$\mathbf{T}_x^{-1}\dot{\mathbf{x}} = \dot{\tilde{\mathbf{x}}}, \quad (10.17)$$

and taking the derivative with respect to time of equation (10.17) yields

$$\mathbf{T}_x^{-1}\dot{\mathbf{x}} = \dot{\tilde{\mathbf{x}}}. \quad (10.18)$$

Substituting equations (10.17) and (10.18) into equation (10.15,a) and rearranging yields equation (10.16,a), where

$$\mathbf{A} = \mathbf{T}_x \tilde{\mathbf{A}} \mathbf{T}_x^{-1} \quad (10.19,a)$$

and

$$\mathbf{B} = \mathbf{T}_x \tilde{\mathbf{B}}. \quad (10.19,b)$$

Following the process outlined above, equation (10.16,b) is obtained, where

$$\mathbf{C} = \mathbf{T}_y \tilde{\mathbf{C}} \mathbf{T}_x^{-1} \quad (10.20,a)$$

and

$$\mathbf{D} = \mathbf{T}_y \tilde{\mathbf{D}}. \quad (10.20,b)$$

10.5 Units

The simulation states and outputs have units of radians for all angular quantities. Therefore, the linear models also have units of radians. For convenience, let the linear models be expressed in units of degrees. To convert the linear models generated from the simulation from units of radians to units of degrees, let equation (10.16) be rewritten as

$$\dot{\mathbf{x}}_r = \mathbf{A}_r \mathbf{x}_r + \mathbf{B}_r \mathbf{u}_d \quad (10.21,a)$$

$$\mathbf{y}_r = \mathbf{C}_r \mathbf{x}_r + \mathbf{D}_r \mathbf{u}_d, \quad (10.21,b)$$

where the symbols ' r ' and ' d ' denote radians and degrees respectively. The state vector in units of degrees is related to the state vector in units of radians by

$$\mathbf{x}_d = \mathbf{S}_x \mathbf{x}_r, \quad (10.22)$$

where

$$\mathbf{S}_x = \left[\begin{array}{c|c} \mathbf{S}_{s_{12 \times 12}} & \mathbf{0}_{12 \times 8} \\ \hline \mathbf{0}_{8 \times 12} & \mathbf{I}_{8 \times 8} \end{array} \right], \quad (10.23)$$

where

$$S_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} \end{bmatrix} \quad (10.24)$$

Leftmultiplying both sides of equation (10.22) by S_x^{-1} gives

$$S_x^{-1} \dot{x}_d = \dot{x}_r \quad (10.25)$$

Taking the time derivative of equation (10.25) yields

$$S_x^{-1} \ddot{x}_d = \ddot{x}_r \quad (10.26)$$

Substituting equations (10.25) and (10.26) into equation (10.21,a) and rearranging gives the following equation:

$$\ddot{x}_d = A_d \dot{x}_d + B_d \ddot{u}_d, \quad (10.27)$$

where

$$A_d = S_x A_r S_x^{-1} \quad (10.28,a)$$

and

$$B_d = S_x B_r \quad (10.28,b)$$

Similarly, the following expression relates the output vector in units of degrees to the output vector in units of radians:

$$y_d = S_y y_r, \quad (10.29)$$

where S_y is given by

$$S_y = \begin{bmatrix} S_{10 \times 10} & 0_{10 \times 8} & 0_{10 \times 2} & 0_{10 \times 11} \\ 0_{8 \times 10} & I_{8 \times 8} & 0_{8 \times 2} & 0_{8 \times 11} \\ 0_{2 \times 10} & 0_{2 \times 8} & S_{2 \times 2} & 0_{2 \times 11} \\ 0_{11 \times 10} & 0_{11 \times 8} & 0_{11 \times 2} & I_{11 \times 11} \end{bmatrix}, \quad (10.30)$$

where

$$S_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{180}{\pi} \end{bmatrix} \quad (10.31)$$

and

$$S_e = \begin{bmatrix} \frac{180}{\pi} & 0 \\ 0 & \frac{180}{\pi} \end{bmatrix}. \quad (10.32)$$

Following the process outlined above, the linear model for the output vector in degrees is given by

$$y_d = C_d x_d + D_d u_d, \quad (10.33)$$

where

$$C_d = S_y C_r S_x^{-1} \quad (10.34,a)$$

and

$$\mathbf{D}_d = \mathbf{S}_y \mathbf{D}_r. \quad (10.34,b)$$

10.6 Linear Model Implementation

MATLAB uses the function *linmod* to extract a linear state-space model of a system about an operating point. *linmod* perturbs each state and each input about the operating point to determine changes in the state derivatives and the outputs. The states are perturbed by

$$x(i) + \Delta x(i), \quad (10.35)$$

where $\Delta x(i)$ is the perturbation associated with the i^{th} state. Similarly, the inputs are perturbed by

$$u(j) + \Delta u(j), \quad (10.36)$$

where $\Delta u(j)$ is the perturbation for the j^{th} input. This method only considers the data on “one side” of the operating point and may produce incorrect linear models. For example, this perturbation method generates incorrect results for the effects of lateral control surfaces on longitudinal state derivatives and outputs. Because of this deficiency in *linmod*, the *linmod* source file was modified to include positive and negative perturbations about an operating point. The states are now perturbed by

$$x(i) \pm \Delta x(i), \quad (10.37)$$

and the inputs are perturbed by

$$u(j) \pm \Delta u(j). \quad (10.38)$$

The function *linearmodel* generates a state-space linear model using the perturbation technique given by equation (10.37) and equation (10.38). The input to *linearmodel* is the operating point which is defined by the state vector, \mathbf{x} , and the input vector, \mathbf{u} . *linearmodel* outputs the state-space linear model given by equation (10.15).

The SIMULINK block diagram used for generating linear models is shown in Figure 10.1. The Dryden Turbulence Model is not included. The lead-lag filter and lag filter in the engine dynamics are implemented in state-space form so that the engine states may be easily extracted. **Please note that any changes to the block diagram may alter the order of the states and cause the generation of incorrect linear models.** As discussed in Section 8.2, the aerodynamic model must be initialized before generating linear models and if the workspace variables are changed during a session the block diagram must be reinitialized, as discussed in Section 8.2. Figure 10.2 gives an example m-file used to set up the linear model problem. The m-file *refh_linearmodel.m* generates a linear model with the desired states and outputs as given by equations (10.27) and (10.33).

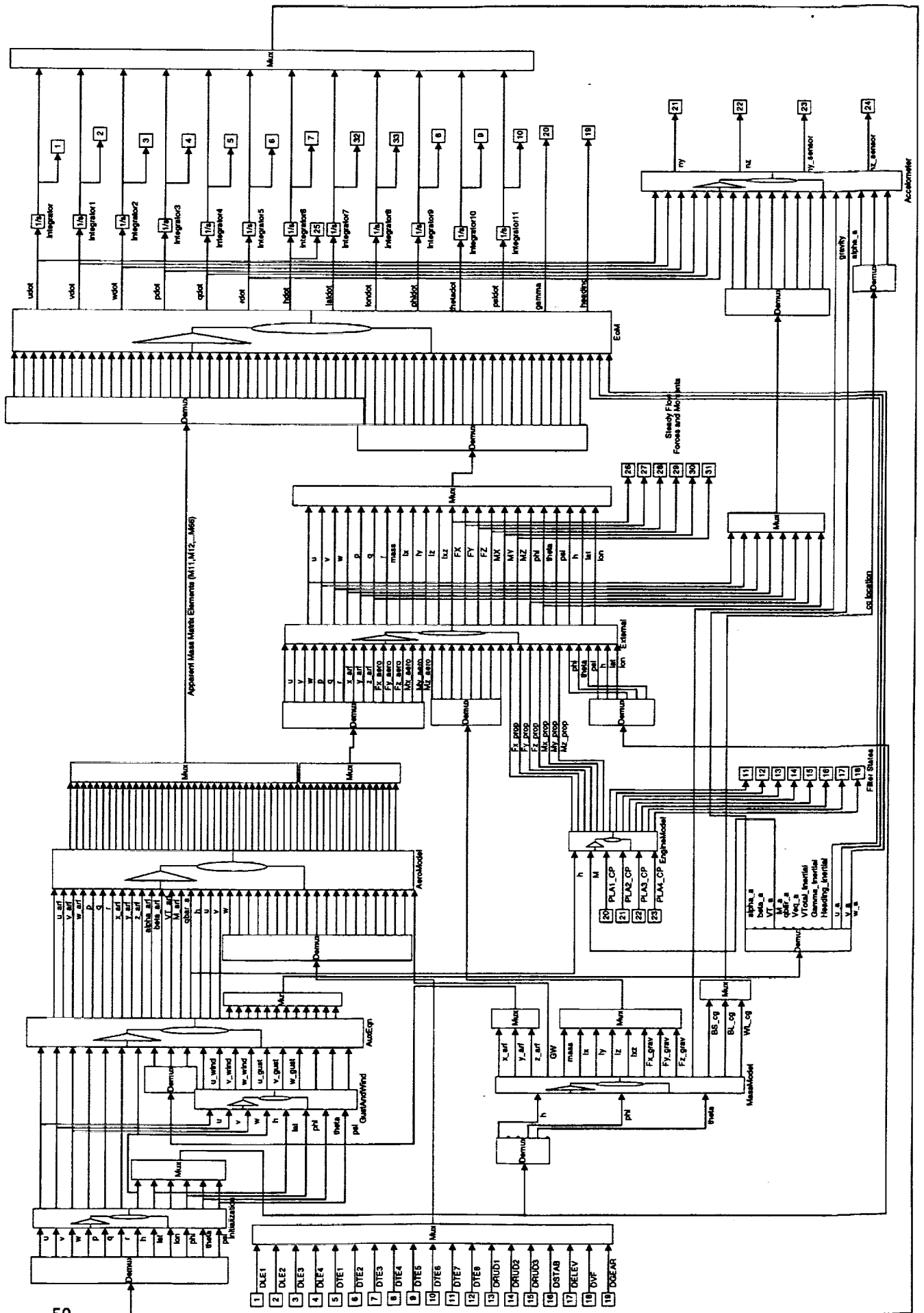


Figure 10.1 Linear Model SIMULINK Implementation – refsim lin.m


```
% GENERATES LINEAR MODEL FOR fc2deg -> Supersonic Cruise Mid
```

```
GW = 0.501324e6;
INIT_FLAG = 0.;
RIGID_FLAG = 0.;
AUTOWT_FLAG = 1.;
AUTO CG_FLAG = 1.;
TLIMIT_FLAG = 1.;
ROTATION_FLAG = 0.;
XCG_PMAC = 54.8146;
YCG_BL_MD = 0.;
ZCG_WL_MD = 203.15;
```

```
% SENSOR LOCATION
```

```
sensor = [
150.;           % BS (in)
-30.;           % BL (in)
-36.;           % WL (in)
];
```

```
% DEFINE INITIAL STATE VECTOR
```

```
x0 = [
% ENGINE STATES
40.;           % wb4 (deg/s)
100.;          % t114 (s)
40.;           % wb3 (deg/s)
100.;          % t113 (s)
40.;           % wb2 (deg/s)
100.;          % t112 (s)
40.;           % wb1 (deg/s)
100.;          % t111 (s)
% SIMULATION STATES
2318.8;        % u (ft/s)
0.;            % v (ft/s)
145.96;        % w (ft/s)
0.;            % p (rad/s)
0.;            % q (rad/s)
0.;            % r (rad/s)
57794.;        % h (ft)
0.*pi/180.;    % lat (rad)
0.*pi/180.;    % lon (rad)
0.*pi/180.;    % phi (rad)
3.6018*pi/180.; % theta (rad)
-90.*pi/180.;  % psi (rad)
];
```

```
% DEFINE INITIAL INPUT VECTOR
```

```
u0 = [
0.0000000000E+00; % DLE1 (deg)
0.0000000000E+00; % DLE2 (deg)
0.0000000000E+00; % DLE3 (deg)
0.0000000000E+00; % DLE4 (deg)
0.0000000000E+00; % DTE1 (deg)
0.0000000000E+00; % DTE2 (deg)
0.0000000000E+00; % DTE3 (deg)
0.0000000000E+00; % DTE4 (deg)
0.0000000000E+00; % DTE5 (deg)
0.0000000000E+00; % DTE6 (deg)
0.0000000000E+00; % DTE7 (deg)
0.0000000000E+00; % DTE8 (deg)
0.0000000000E+00; % DRUD1 (deg)
0.0000000000E+00; % DRUD2 (deg)
0.0000000000E+00; % DRUD3 (deg)
];
```

```

0.1892955855E+01;      % DSTAB (deg)
0.3785911710E+01;      % DELEV (deg)
0.0000000000E+00;      % DVF (deg)
0.0000000000E+00;      % DGEAR (deg)
0.1000000000E+03;      % PLA1_CP (deg)
0.1000000000E+03;      % PLA2_CP (deg)
0.1000000000E+03;      % PLA3_CP (deg)
0.1000000000E+03;      % PLA4_CP (deg)
];

% STATE-SPACE REPRESENTATION FOR LEAD-LAG FILTER
ATau = [-4.];
BTau = [1.];
CTau = [-.33333; 1.];
DTau = [.33333; 0];

% STATE-SPACE REPRESENTATION FOR LAG FILTER
AOmega = [-2.5];
BOmega = [1.];
COmega = [2.5; 1.];
DOmega = [0.; 0.];

```

Figure 10.2 Example M-file To Setup The Linear Model Problem - fc2deg_setup.m

10.7 Linear Model Results

Time history results are shown in Figure 10.3 for the linear model generated at the flight condition defined in Figure 10.2, with a one degree doublet applied to the horizontal stabilizer and a corresponding two degree doublet applied to the elevator. Atmospheric disturbances are neglected. Results generated by the MATLAB/SIMULINK nonlinear simulation are also presented for comparison. Both sets of time histories were generated at 40 Hz.

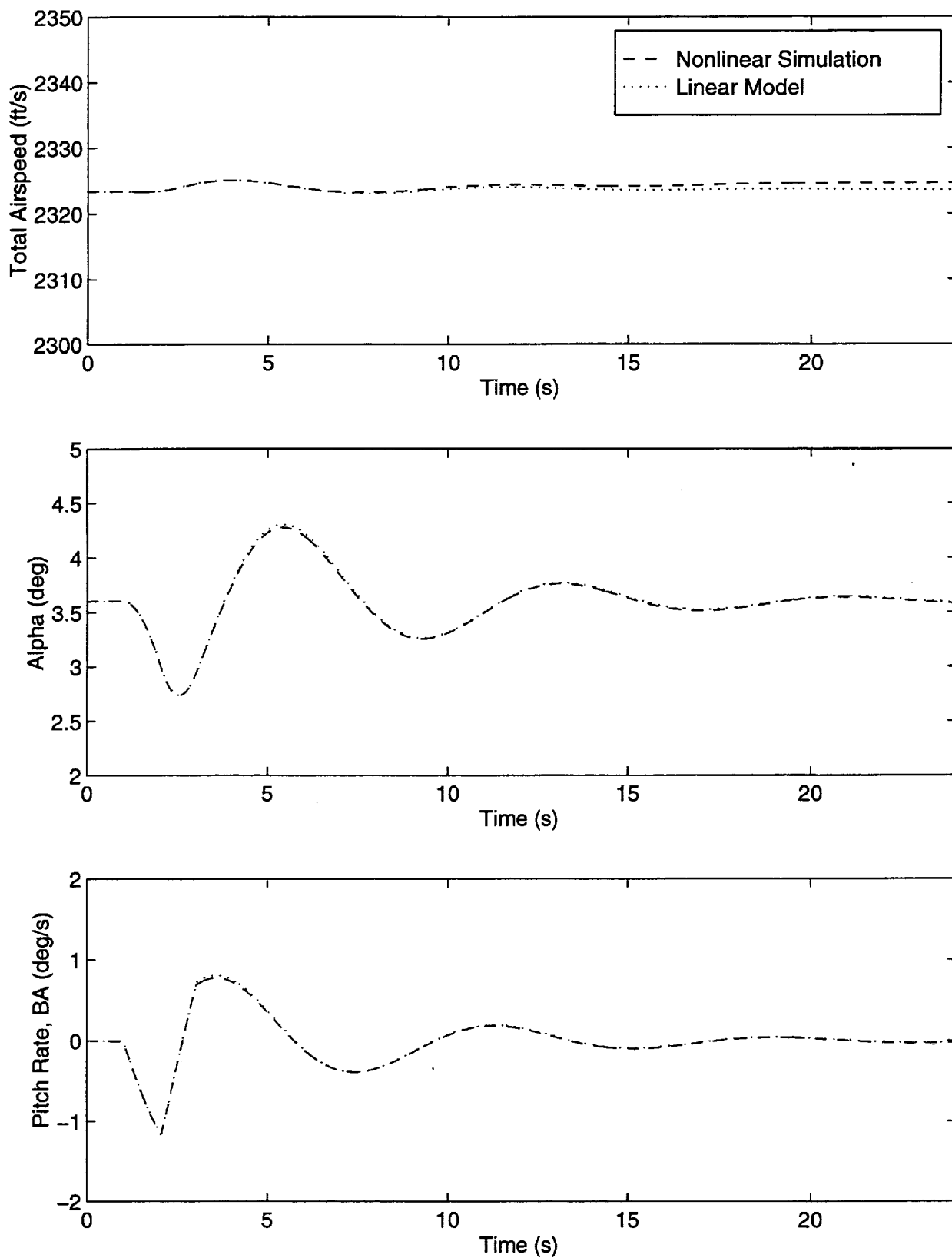


Figure 10.3 Linear Model And Nonlinear Simulation Dynamic Response To A Stabilizer Doublet

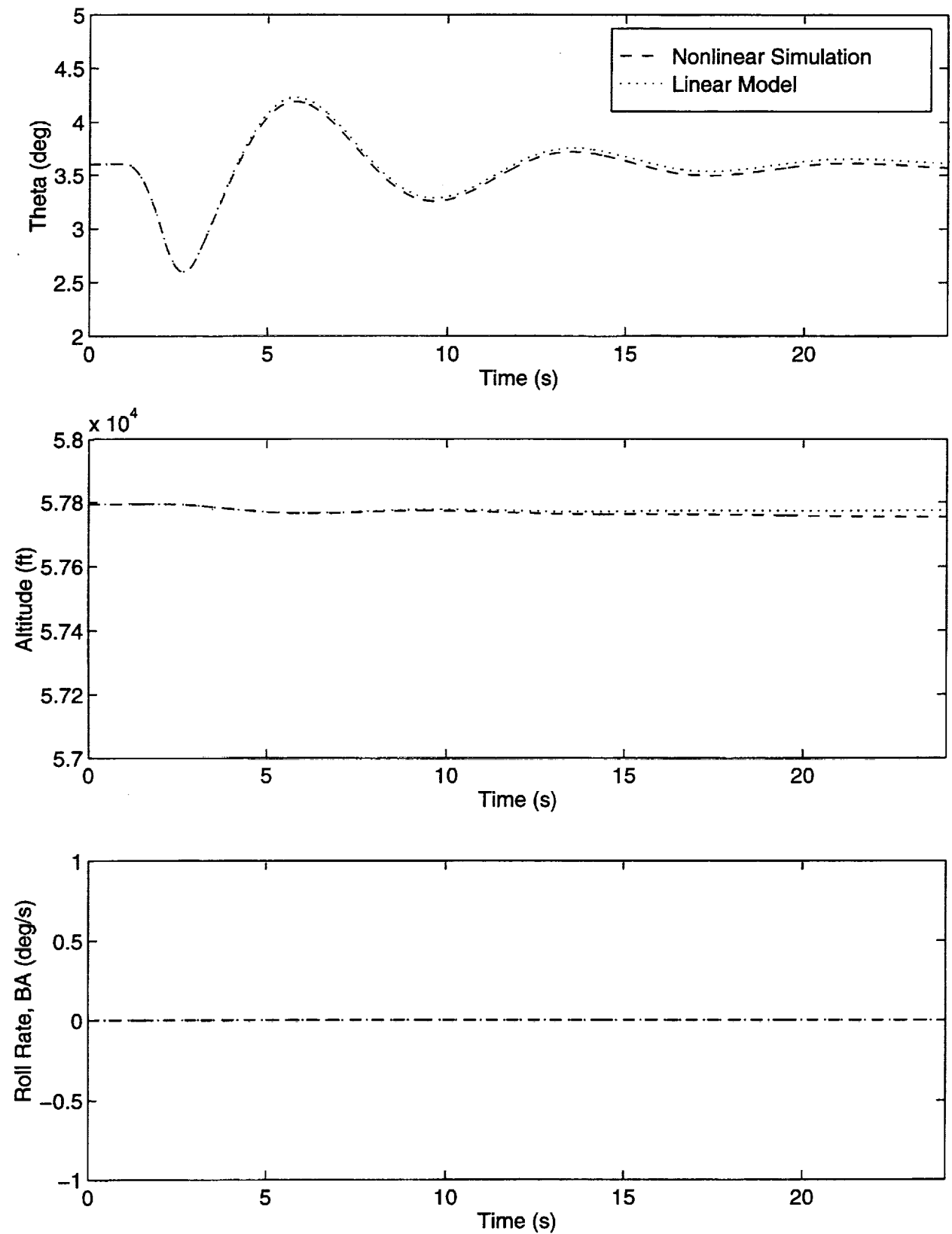


Figure 10.3 (Continued)

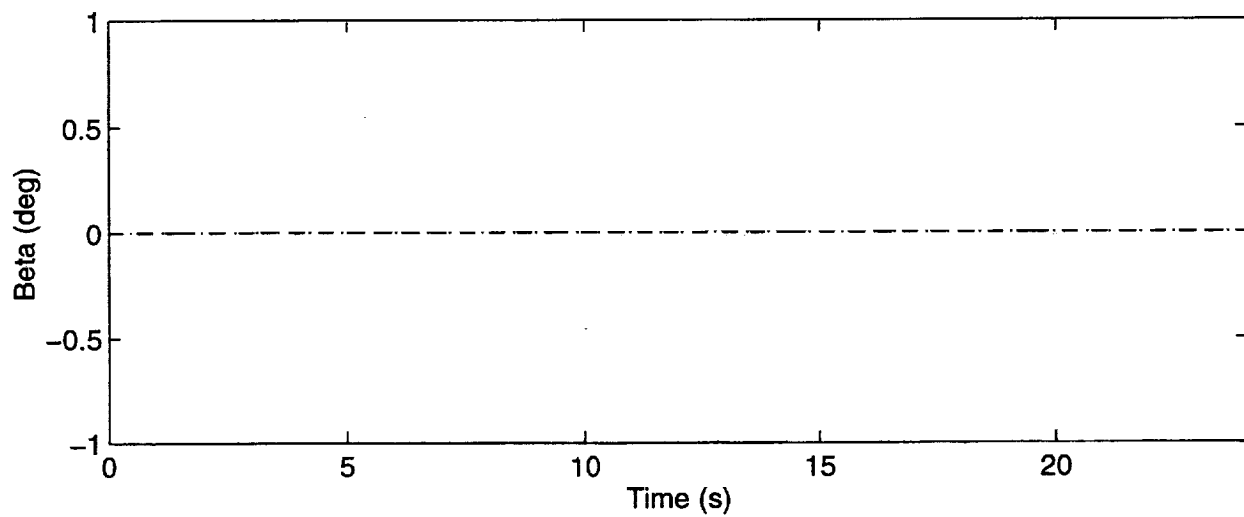
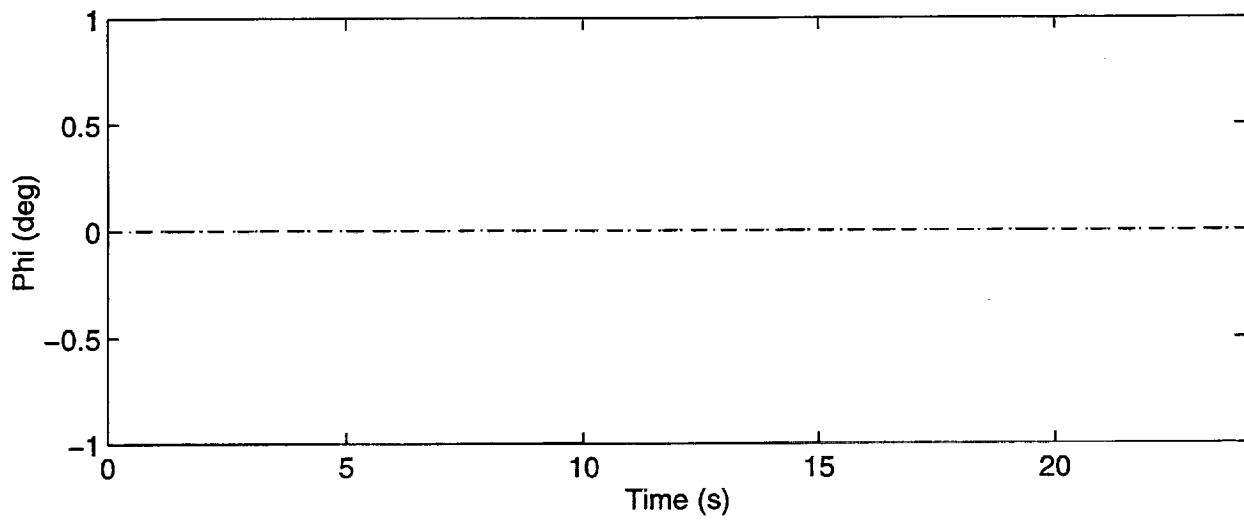
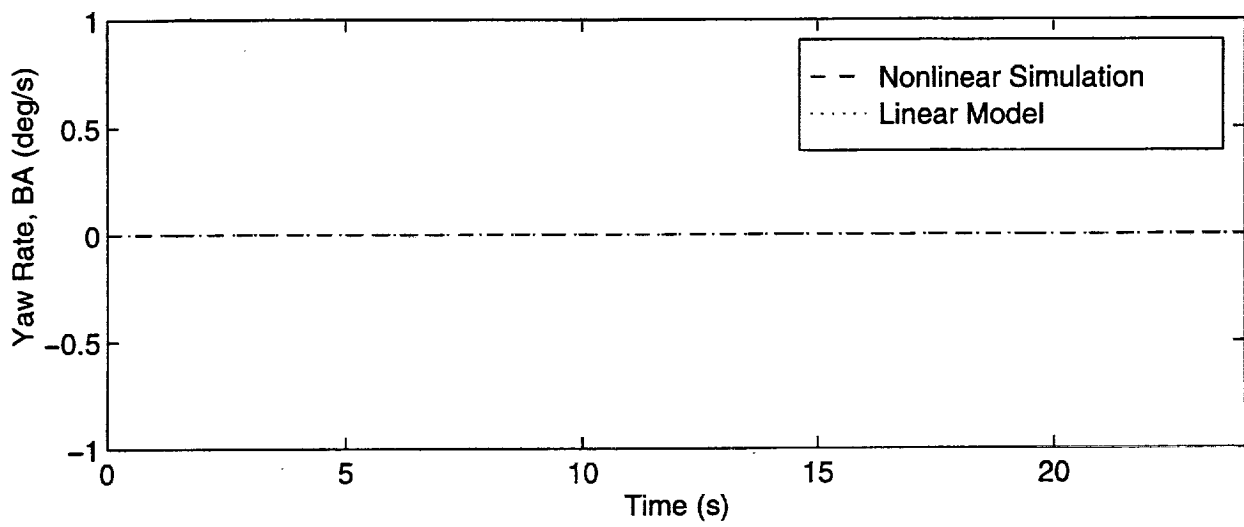


Figure 10.3 (Continued)

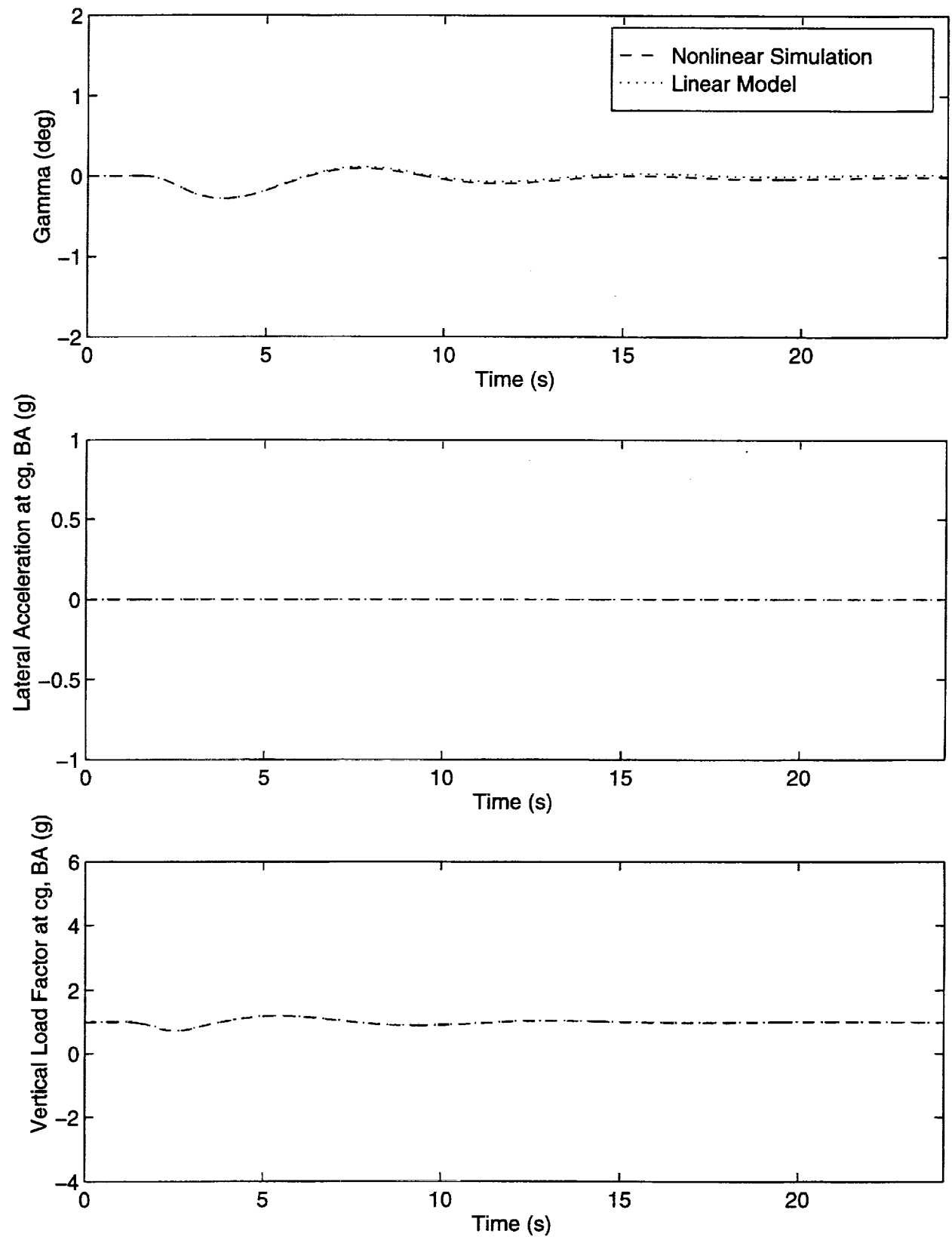


Figure 10.3 (Continued)

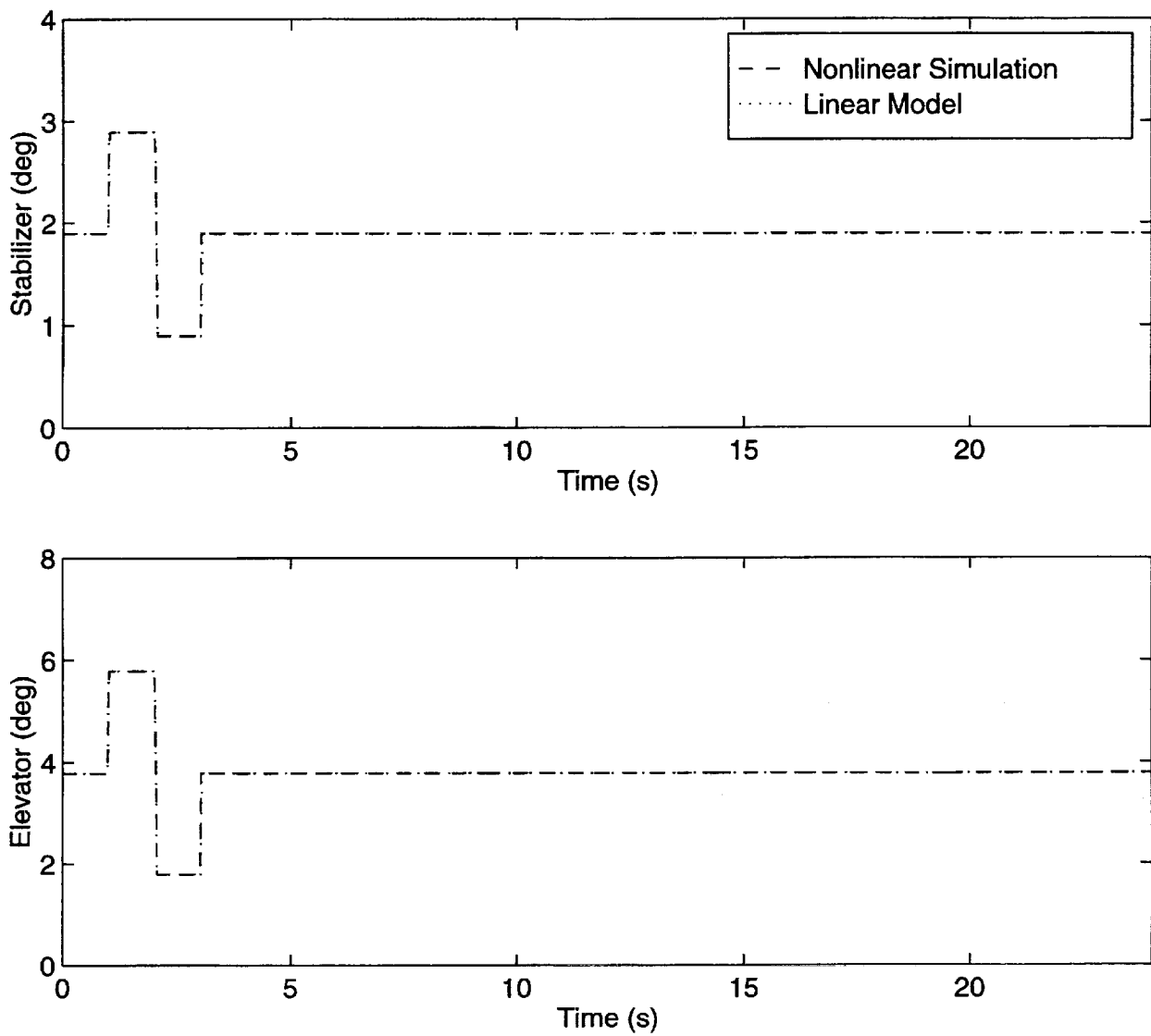


Figure 10.3 (Concluded)

11 Concluding Remarks

The mathematical model and associated code used to simulate the Boeing Reference H configuration high speed civil transport aircraft are described in this report. The simulation is implemented using MATLAB/SIMULINK software, a commercially available control system engineering tool, to support advanced control law research. The simulation software includes the capability to calculate trim solutions and to generate linear models at specified flight conditions.

The simulation math model includes the nonlinear, six degree-of-freedom equations which govern the motion of a rigid aircraft in atmospheric flight. The Earth atmosphere is modeled with the 1962 Standard Atmosphere and a Dryden turbulence model. The Reference H configuration is modeled using the Boeing Cycle 1 simulation data base, which consists of geometry information, tabular aerodynamic model, an engine model, and a mass model. Models of actuator dynamics, landing gear, and flight control system are not implemented in the simulation.

To validate the MATLAB/SIMULINK simulation software, a comparison of simulation-generated dynamic responses with those from independent simulations at Boeing Commercial Aircraft Company and NASA Langley Research Center's VMS were made. A good match of the responses was achieved for the dynamic check cases provided. The trim solutions obtained with the developed software compared closely to the trim check cases provided by Boeing. The linear models obtained at representative flight conditions compared closely to those generated using the NASA Langley Research Center's real time simulation used for the Visual Motion Simulation. Furthermore, time histories generated using the linear model compared closely to those generated by the nonlinear simulation.

The source code is available from the authors.

References

- Buttrill, C. S., Arbuckle, P. D., and Hoffler, K. D.; "Simulation Model of a Twin-Tail, High Performance Airplane," NASA Technical Memorandum 107601, July 1992.
- Dawdy, J. R., Dornfield, G. M., and Phillips, B. A.; *High Speed Civil Transport Reference H - Cycle 1 Simulation Data Base*, NASA Contract NAS1-19360 Task Assignment No. 49, June 1994.
- Etkin, B.; *Dynamics of Atmospheric Flight*; John Wiley & Sons, Inc.; New York, 1972.
- Etkin, B.; *Dynamics of Flight Stability And Control*, John Wiley & Sons, Inc.; New York, 1982.
- Grace, A.; *Optimization Toolbox User's Guide*, The MathWorks, Inc.; November 1992.
- Hoblitt, F. M.; *Gust Loads on Aircraft: Concepts and Applications*, AIAA Education Series, AIAA Inc., 1988.
- Jackson, E. B.; "Manual for a Workstation-based Generic Flight Simulation Program (LaRCsim) Version 1.4," NASA Technical Memorandum 110164, May 1995.
- Johnson, D. L.; Editor: *Terrestrial Environment (Climatic) Criteria for Use in Aerospace Vehicle Development, 1993 Revision*, NASA Technical Memorandum 4511, August 1993.
- MATLAB High Performance Numeric Computation and Visualization Software User's Guide*, The MathWorks, Inc.; August 1992.
- McFarland, R. E.; "A Standard Kinematic Model For Flight Simulation At NASA Ames," NASA Contract Report 2497, January 1975.
- SIMULINK Dynamic System Simulation Software User's Guide*, The MathWorks, Inc.; December 1993.
- Stephens, A. T., Dornfield, G. M., Kuta, J. F., Lanier, J. K., Milligan, K. H., and Phillips, B. A.; *High Speed Civil Transport Reference H - Cycle 2A Simulation Data Base*, NASA Contract NAS1-20220 Task Assignment No. 7, March 1995.
- Stephens, A. T., Dornfield, G. M., Lanier, J. K., Milligan, K. H., Parker, J. M., and Phillips, B. A.; *High Speed Civil Transport Reference H - Cycle 2B Simulation Data Base*, NASA Contract NAS1-20220 Task Assignment No. 7, July 1995.

Symbols and Abbreviations

Acronyms

arc	aerodynamic reference center
BA	body axis reference frame
BL	location along wing buttock line, in
BS	location along body station, in
cg	center of gravity
SA	stability axis reference frame
WL	location along waterline, in

Symbol	Description
A, B, C, D	state-space matrices in the linear model description
a	speed of sound, ft/s
b	wing span reference length, ft
C_D	drag coefficient
C_L	lift coefficient
C_l	rolling moment coefficient
C_m	pitching moment coefficient
C_n	yawing moment coefficient
C_Y	sideforce coefficient
\bar{c}	wing chord reference length, ft
$\{F\}$	generalized force vector
$\{F\}_I$	generalized inertial force vector
F	force, lbf
$F_u(s)$	longitudinal turbulence frequency response function
$F_w(s)$	lateral-vertical turbulence frequency response function
G	trim constraint vector
g	gravitational acceleration, ft/s ²
g_0	gravitational acceleration at sea level, ft/s ²
h	altitude above sea level, ft
I_x	roll moment of inertia, slug-ft ²
I_{xz}	roll-yaw product of inertia, slug-ft ²
I_y	pitch moment of inertia, slug-ft ²
I_z	yaw moment of inertia, slug-ft ²
L_u	longitudinal turbulence spatial scale length, ft
L_w	lateral-vertical turbulence spatial scale length, ft
$[M]_A$	apparent mass matrix
$[M]_I$	inertial mass matrix
M	Mach number or moment, ft-lbf
m	mass, slugs
n_x	axial accelerometer output, g units
n_y	lateral accelerometer output, g units

n_z	vertical accelerometer output, g units
\dot{P}	x-body frame component of rotational velocity of the body reference frame relative to the local reference frame, rad/s
\dot{p}	x-body frame component of rotational velocity, rad/s
\dot{Q}	y-body frame component of rotational velocity of the body reference frame relative to the local reference frame, rad/s
\dot{q}	y-body frame component of rotational velocity, rad/s
\bar{q}	dynamic pressure, lbf/ft ²
R	distance from center of Earth to aircraft mass center, ft or z-body frame component of rotational velocity of the body reference frame relative to the local reference frame, rad/s
R_E	Earth radius, ft
\dot{r}	z-body frame component of rotational velocity, rad/s
S	wing reference area, ft ²
S_e	sub-matrix of the output units transformation matrix
S_s	sub-matrix of the state and output units transformation matrix
S_x	state units transformation matrix
S_y	output units transformation matrix
$[T_{BV}]$	transformation matrix from vehicle-carried local horizontal reference frame to body reference frame
$[T_{BS}]$	transformation matrix from stability reference frame to body reference frame
T_e	sub-matrix of the state vector transformation matrix
T_o	sub-matrix of the output vector transformation matrix
T_s	sub-matrix of the state vector transformation matrix
T_x	state vector transformation matrix
T_y	output vector transformation matrix
t	time, s
\mathbf{u}	control vector
u	x-body frame component of inertial translational velocity, ft/s
$\{V\}$	generalized velocity vector
V_D	downward velocity component, ft/s
V_E	eastward velocity component, ft/s
V_{Ec}	eastward velocity component corrected for Earth rotation, ft/s
V_N	northward velocity component, ft/s
V_T	total airspeed, ft/s
v	y-body frame component of inertial translational velocity, ft/s
w	z-body frame component of inertial translational velocity, ft/s
\mathbf{X}	trim input vector
\mathbf{x}	state vector
x_{arf}	distance from cg to arc along x-body axis, + forward, ft
x_s	distance from cg to sensor along x-body axis, + forward, ft
\mathbf{y}	output vector
y_{arf}	distance from cg to arc along y-body axis, + right, ft
y_s	distance from cg to sensor along y-body axis, + right, ft
z_{arf}	distance from cg to arc along z-body axis, + down, ft
z_s	distance from cg to sensor along z-body axis, + down, ft

α	angle of attack, deg or rad
β	sideslip angle, deg or rad
Δt	integration time step, s
δ_e	elevator deflection, deg
δ_{pla}	power lever angle, deg
δ_{stab}	stabilizer deflection, deg
ϕ	Euler roll angle, rad
γ_h	horizontal flight path angle measured from north, rad
γ_v	vertical flight path angle, rad
λ	latitude, rad
θ	Euler pitch angle, rad
ρ	air density, slug/ft ³
σ_u	variance of longitudinal turbulence, ft/s
σ_w	variance of lateral-vertical turbulence, ft/s
τ	longitude, rad
τ_{LL}	engine lead-lag filter state, s
$[\Omega]$	body frame inertial rotation matrix
ω_B	engine lag filter state, deg/s
ω_E	Earth rotation, rad/s
ψ	Euler yaw angle, rad

Subscripts and Superscripts

0	initial
<i>Aero</i>	due to air loads
<i>a</i>	aerodynamic, computed at center of gravity with respect to air mass
<i>airmass</i>	due to winds, gusts, and turbulence
<i>arf</i>	computed at aerodynamic reference center
<i>b</i>	body reference frame
<i>E</i>	Earth
<i>d</i>	degrees
<i>Grav</i>	due to gravity
<i>Prop</i>	thrust-induced
<i>ps</i>	pilot station
<i>r</i>	radians
<i>s</i>	sensor
<i>sf</i>	steady flow
<i>usf</i>	unsteady flow
<i>u</i>	longitudinal atmospheric turbulence quantities
<i>w</i>	lateral-vertical atmospheric turbulence quantities

Simulation Files

Function Files

File	Description
Accel.m	performs accelerometer output calculations
AeroCoeff.m	transfers steady flow aerodynamic force and moment coefficients from the aerodynamic reference center to the center of gravity
AuxEqn.m	performs auxiliary calculations which drive the aircraft model and the equations of motion
DrydenVel.m	calculates airspeed for use in the Dryden turbulence model
dryden_model_coeff.m	calculates turbulence frequency response function coefficients
Eom.m	calculates the state derivative vector and auxiliary quantities
External.m	calculates total external forces and moments acting at the center of gravity
GravForce.m	calculates gravitational force components
linearmodel.m	generates linear model in state-space form
longtrim_fun.m	trim optimization function

S-function MEX Files

atmosphere_62.mex4	1962 Standard Atmosphere model
refhaero.mex4	Reference H cycle 1 aerodynamic model
refhengine.mex4	Reference H cycle 1 engine model
refhmass.mex4	Reference H cycle 1 mass model

SIMULINK Files

refhsim.m	nonlinear simulation block diagram
refhsim_lin.m	linear model block diagram
refhsim_trim.m	trim solution block diagram

Auxiliary M-Files

aero_init.m	initializes aerodynamic model
fc2deg_setup.m	sets up flight condition and simulation to compute linear model at the desired flight condition.
longtrim.m	sets up and finds longitudinal trim solution
plot_var.m	generates plotting variables from simulation output
refh_linearmodel.m	first computes the linear model in body axis states and then transforms linear model to desired states and output
yawdoub_setup.m	sets initial flight condition and simulation variables for rudder doublet dynamic response

Appendix A - Apparent Mass Matrix Elements

Each element of the 6x6 apparent mass matrix is defined below. Given the following two intermediate terms:

$$m_0 = \bar{q}S \left(\frac{\bar{c}}{2V_T} \right) \left(\frac{u_{arf}}{u_{arf}^2 + w_{arf}^2} \right) \quad (A.1)$$

$$m_1 = \bar{q}S \left(\frac{b}{2V_T} \right) \left(\frac{1}{\sqrt{u_{arf}^2 + w_{arf}^2}} \right). \quad (A.2)$$

The apparent mass matrix elements are given by

$$\frac{\partial F_{X_{usf}}}{\partial \dot{u}_a} = -m_0 C_{X\dot{\alpha}_{arf}} \frac{w_{arf}}{u_{arf}} \quad (A.3)$$

$$\frac{\partial F_{X_{usf}}}{\partial \dot{v}_a} = 0 \quad (A.4)$$

$$\frac{\partial F_{X_{usf}}}{\partial \dot{w}_a} = m_0 C_{X\dot{\alpha}_{arf}} \quad (A.5)$$

$$\frac{\partial F_{Y_{usf}}}{\partial \dot{u}_a} = 0 \quad (A.6)$$

$$\frac{\partial F_{Y_{usf}}}{\partial \dot{v}_a} = m_1 C_{Y\dot{\beta}_{arf}} \quad (A.7)$$

$$\frac{\partial F_{Y_{usf}}}{\partial \dot{w}_a} = 0 \quad (A.8)$$

$$\frac{\partial F_{Z_{usf}}}{\partial \dot{u}_a} = -m_0 C_{Z\dot{\alpha}_{arf}} \left(\frac{w_{arf}}{u_{arf}} \right) \quad (A.9)$$

$$\frac{\partial F_{Z_{usf}}}{\partial \dot{v}_a} = 0 \quad (A.10)$$

$$\frac{\partial F_{Z_{usf}}}{\partial \dot{w}_a} = m_0 C_{Z\dot{\alpha}_{arf}} \quad (A.11)$$

$$\frac{\partial F_{X_{usf}}}{\partial \dot{p}} = m_0 C_{X_{\dot{\alpha}_{arf}}} y_{arf} \quad (\text{A.12})$$

$$\frac{\partial F_{X_{usf}}}{\partial \dot{q}} = -m_0 C_{X_{\dot{\alpha}_{arf}}} \left(x_{arf} + \frac{w_{arf}}{u_{arf}} z_{arf} \right) + \bar{q} S C_{X_{\dot{q}}} \left(\frac{\bar{c}}{2V_T} \right)^2 \quad (\text{A.13})$$

$$\frac{\partial F_{X_{usf}}}{\partial \dot{r}} = m_0 C_{X_{\dot{\alpha}_{arf}}} \left(\frac{w_{arf}}{u_{arf}} y_{arf} \right) \quad (\text{A.14})$$

$$\frac{\partial F_{Y_{usf}}}{\partial \dot{p}} = -m_1 C_{Y_{\dot{\beta}_{arf}}} z_{arf} + \bar{q} S C_{Y_{\dot{p}}} \left(\frac{b}{2V_T} \right)^2 \quad (\text{A.15})$$

$$\frac{\partial F_{Y_{usf}}}{\partial \dot{q}} = 0 \quad (\text{A.16})$$

$$\frac{\partial F_{Y_{usf}}}{\partial \dot{r}} = m_1 C_{Y_{\dot{\beta}_{arf}}} x_{arf} + \bar{q} S C_{Y_{\dot{r}}} \left(\frac{b}{2V_T} \right)^2 \quad (\text{A.17})$$

$$\frac{\partial F_{Z_{usf}}}{\partial \dot{p}} = m_o C_{Z_{\dot{\alpha}_{arf}}} y_{arf} \quad (\text{A.18})$$

$$\frac{\partial F_{Z_{usf}}}{\partial \dot{q}} = -m_o C_{Z_{\dot{\alpha}_{arf}}} \left(x_{arf} + \frac{w_{arf}}{u_{arf}} z_{arf} \right) + \bar{q} S C_{Z_{\dot{q}}} \left(\frac{\bar{c}}{2V_T} \right)^2 \quad (\text{A.19})$$

$$\frac{\partial F_{Z_{usf}}}{\partial \dot{r}} = m_o C_{Z_{\dot{\alpha}_{arf}}} \left(\frac{w_{arf}}{u_{arf}} y_{arf} \right) \quad (\text{A.20})$$

$$\frac{\partial M_{X_{usf}}}{\partial \dot{u}_a} = y_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{u}_a} \quad (\text{A.21})$$

$$\frac{\partial M_{X_{usf}}}{\partial \dot{v}_a} = m_1 b C_{l_{\dot{\beta}_{arf}}} - z_{arf} \frac{\partial F_{Y_{usf}}}{\partial \dot{v}_a} \quad (\text{A.22})$$

$$\frac{\partial M_{X_{usf}}}{\partial \dot{w}_a} = y_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{w}_a} \quad (\text{A.23})$$

$$\frac{\partial M_{Y_{usf}}}{\partial \dot{u}_a} = -m_0 \bar{c} C_{m_{\dot{\alpha}_{arf}}} \left(\frac{w_{arf}}{u_{arf}} \right) + z_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{u}_a} - x_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{u}_a} \quad (\text{A.24})$$

$$\frac{\partial M_{Y_{usf}}}{\partial \dot{v}_a} = 0 \quad (\text{A.25})$$

$$\frac{\partial M_{Y_{usf}}}{\partial \dot{w}_a} = m_0 \bar{c} C_{m_{\dot{\alpha}_{arf}}} + z_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{w}_a} - x_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{w}_a} \quad (\text{A.26})$$

$$\frac{\partial M_{Z_{usf}}}{\partial \dot{u}_a} = -y_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{u}_a} \quad (\text{A.27})$$

$$\frac{\partial M_{Z_{usf}}}{\partial \dot{v}_a} = m_1 b C_{n_{\dot{\beta}_{arf}}} + x_{arf} \frac{\partial F_{Y_{usf}}}{\partial \dot{v}_a} \quad (\text{A.28})$$

$$\frac{\partial M_{Z_{usf}}}{\partial \dot{w}_a} = -y_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{w}_a} \quad (\text{A.29})$$

$$\frac{\partial M_{X_{usf}}}{\partial \dot{p}} = -m_1 b C_{l_{\dot{\beta}_{arf}}} z_{arf} + \bar{q} S b \left(\frac{b}{2V_T} \right)^2 C_{l_p} - z_{arf} \frac{\partial F_{Y_{usf}}}{\partial \dot{p}} + y_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{p}} \quad (\text{A.30})$$

$$\frac{\partial M_{X_{usf}}}{\partial \dot{q}} = y_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{q}} \quad (\text{A.31})$$

$$\frac{\partial M_{X_{usf}}}{\partial \dot{r}} = m_1 b x_{arf} C_{l_{\dot{\beta}_{arf}}} + \bar{q} S b \left(\frac{b}{2V_T} \right)^2 C_{l_r} - z_{arf} \frac{\partial F_{Y_{usf}}}{\partial \dot{r}} + y_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{r}} \quad (\text{A.32})$$

$$\frac{\partial M_{Y_{usf}}}{\partial \dot{p}} = m_0 \bar{c} C_{m_{\dot{\alpha}_{arf}}} y_{arf} + z_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{p}} - x_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{p}} \quad (\text{A.33})$$

$$\frac{\partial M_{Y_{usf}}}{\partial \dot{q}} = -m_0 \bar{c} C_{m_{\dot{\alpha}_{arf}}} \left(x_{arf} + \frac{w_{arf}}{u_{arf}} z_{arf} \right) + \bar{q} S \bar{c} \left(\frac{\bar{c}}{2V_T} \right)^2 C_{m_{\dot{q}}} + z_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{q}} - x_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{q}} \quad (\text{A.34})$$

$$\frac{\partial M_{Y_{usf}}}{\partial \dot{r}} = m_0 \bar{c} C_{m_{\dot{\alpha}_{arf}}} \left(\frac{w_{arf}}{u_{arf}} y_{arf} \right) + z_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{r}} - x_{arf} \frac{\partial F_{Z_{usf}}}{\partial \dot{r}} \quad (\text{A.35})$$

$$\frac{\partial M_{Z_{usf}}}{\partial \dot{p}} = -m_1 b z_{arf} C_{n_{\dot{\beta}_{arf}}} + \bar{q} S b \left(\frac{b}{2V_T} \right)^2 C_{n_p} - y_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{p}} + x_{arf} \frac{\partial F_{Y_{usf}}}{\partial \dot{p}} \quad (\text{A.36})$$

$$\frac{\partial M_{Z_{usf}}}{\partial \dot{q}} = -y_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{q}} \quad (\text{A.37})$$

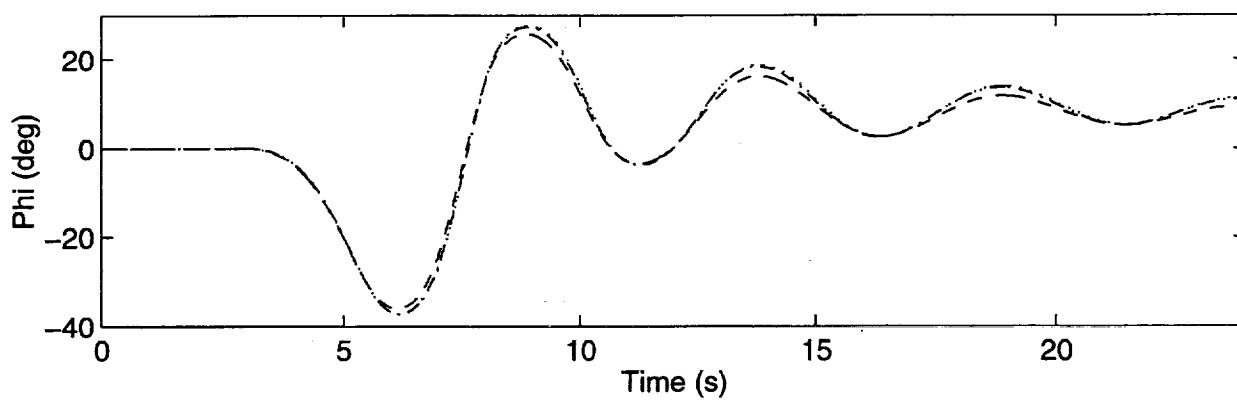
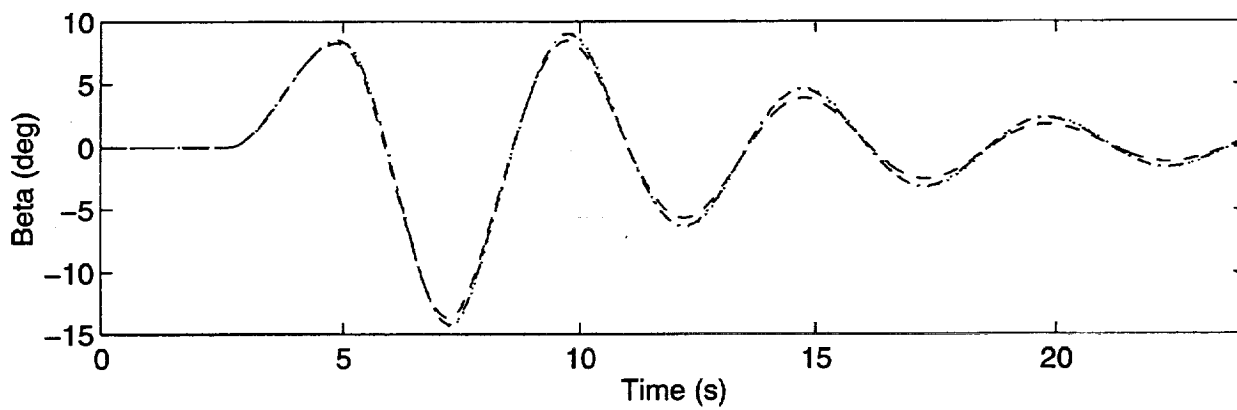
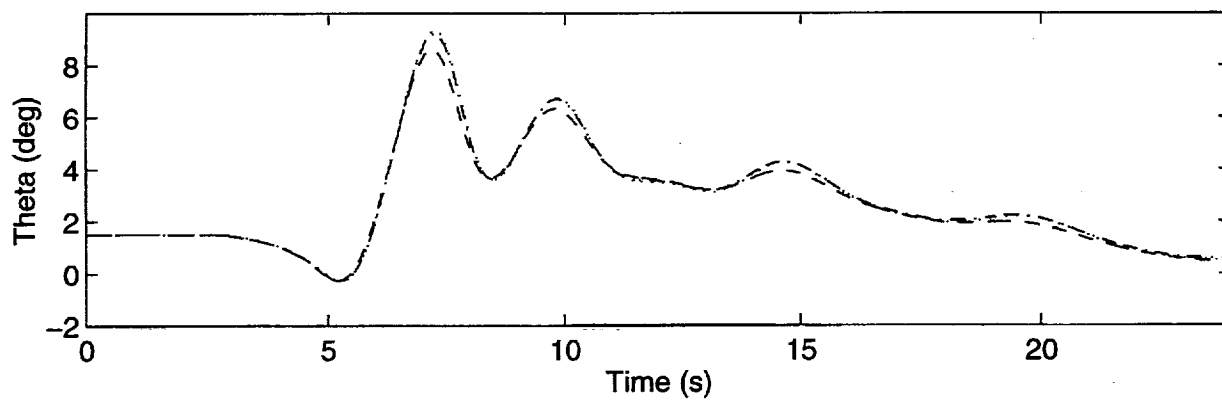
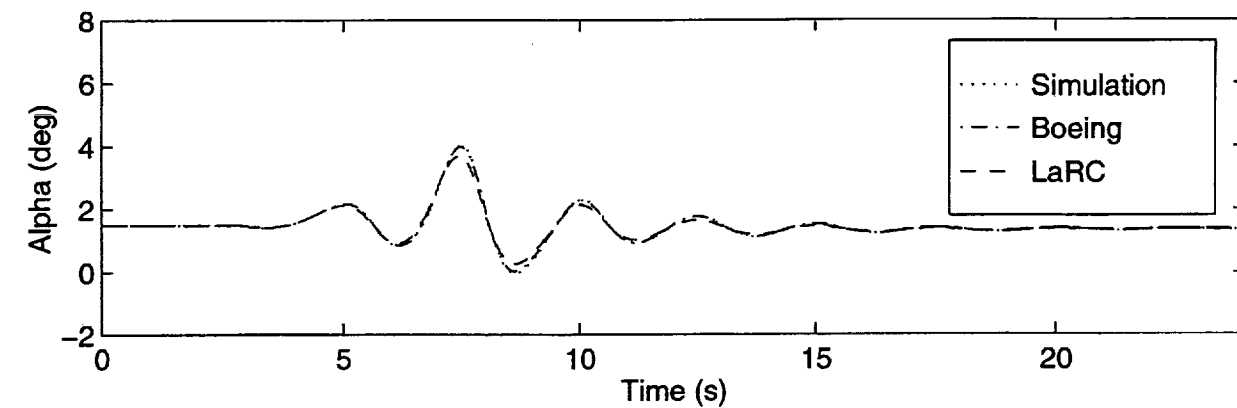
$$\frac{\partial M_{Z_{usf}}}{\partial \dot{r}} = m_1 b x_{arf} C_{n_{\beta_{arf}}} + \bar{q} S b \left(\frac{b}{2V_T} \right)^2 C_{n_{\dot{r}}} - y_{arf} \frac{\partial F_{X_{usf}}}{\partial \dot{r}} + x_{arf} \frac{\partial F_{Y_{usf}}}{\partial \dot{r}} \quad (\text{A.38})$$

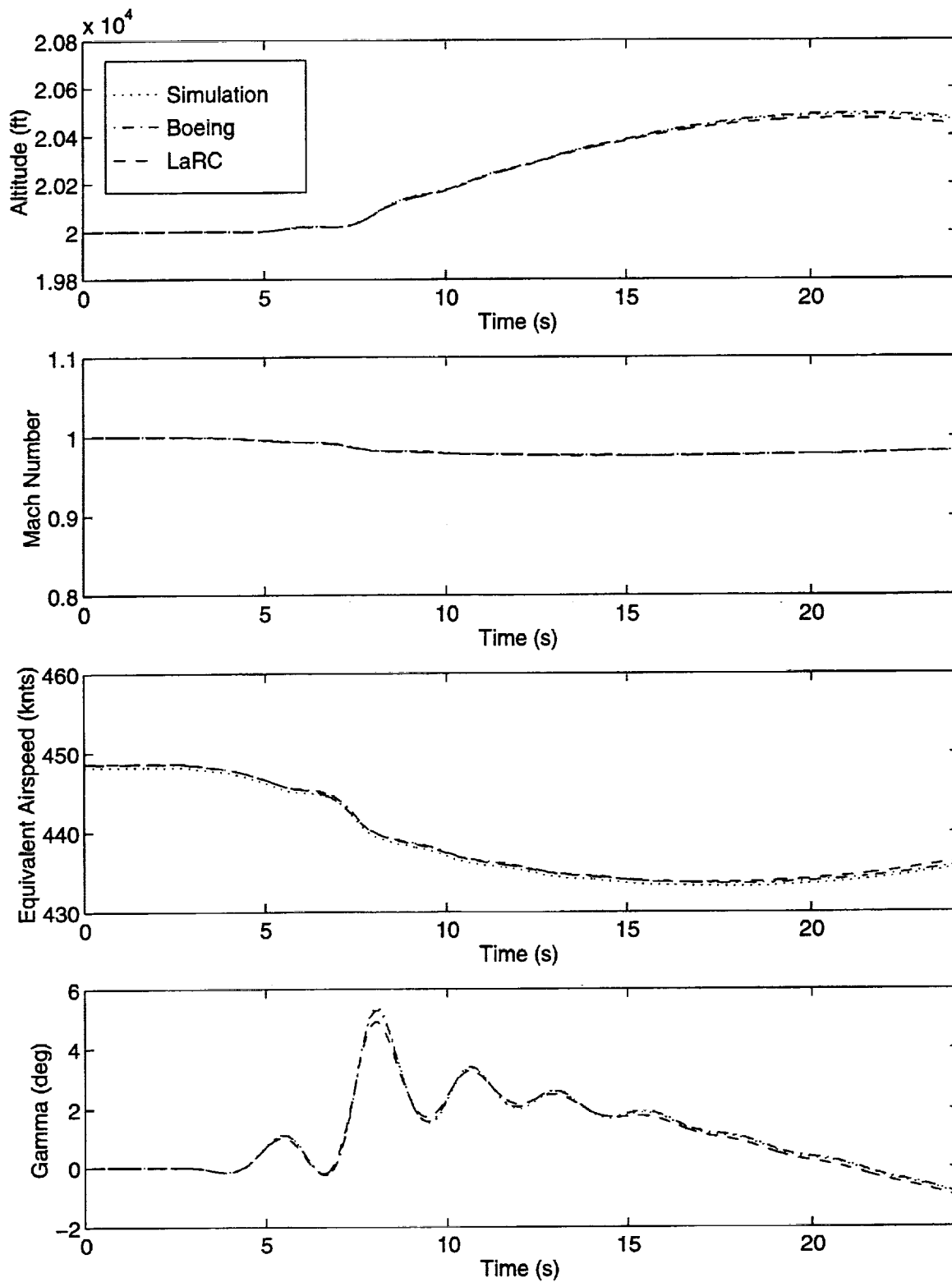
The elements given by equations (A.3) to (A.11) have units of slugs. Elements (A.12) to (A.29) have units of slug-ft and elements (A.30) to (A.38) have units of slug-ft².

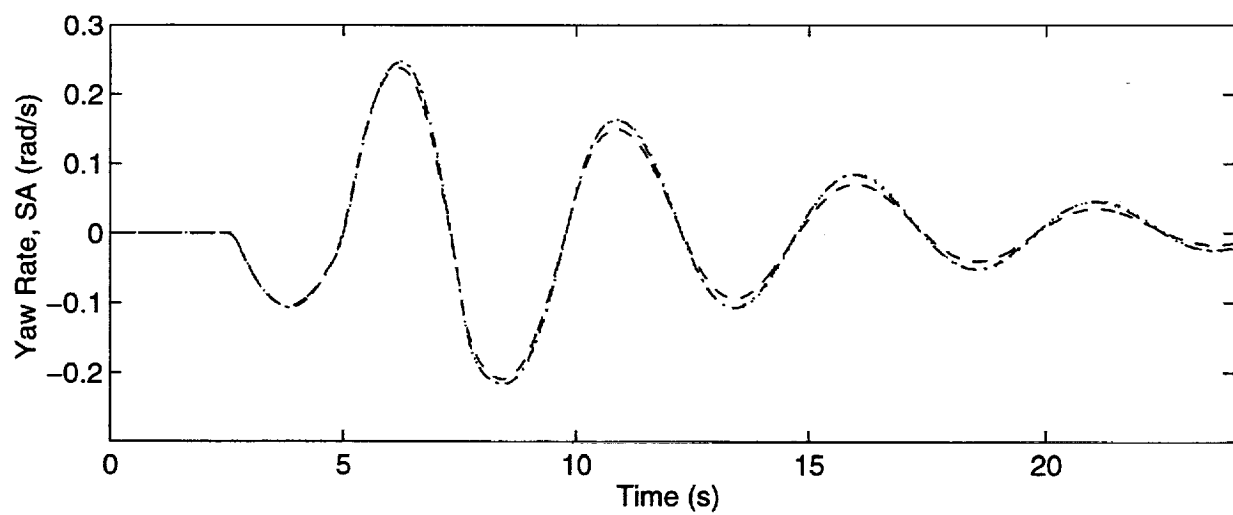
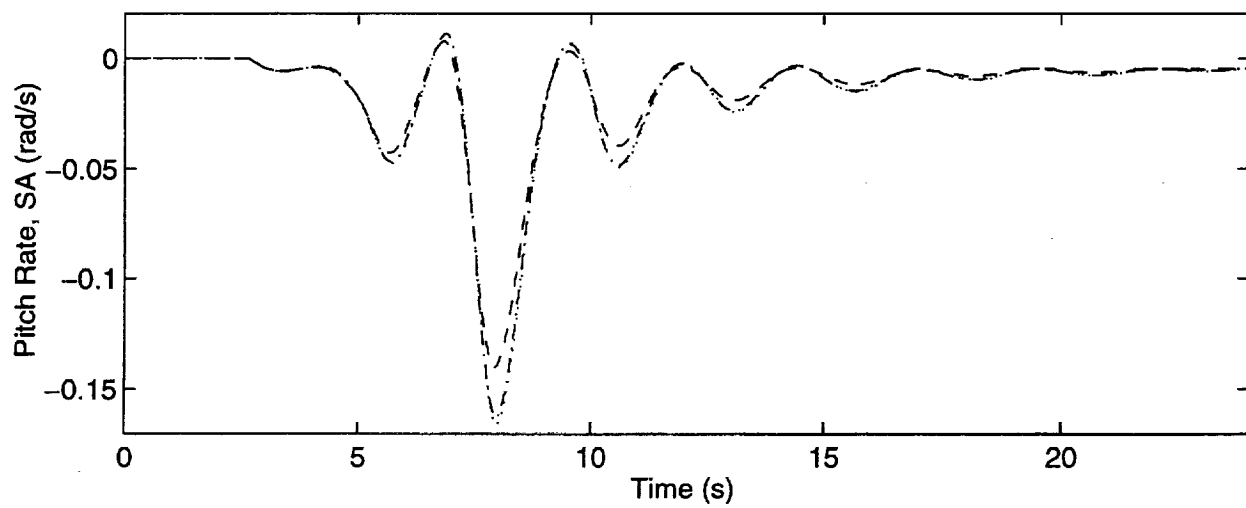
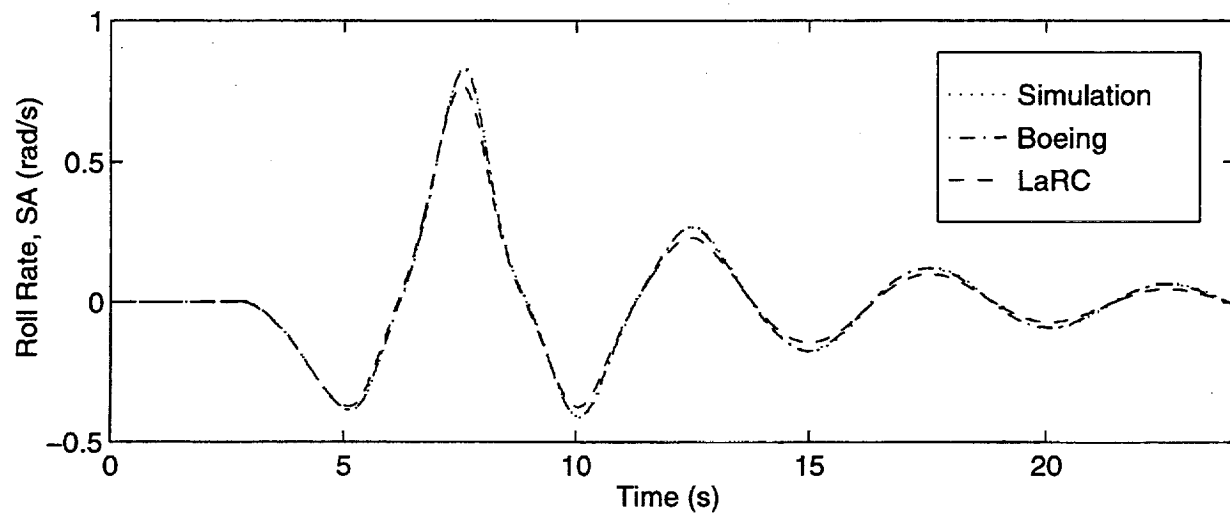
Appendix B - Dynamic Response Comparison To A Rudder Doublet

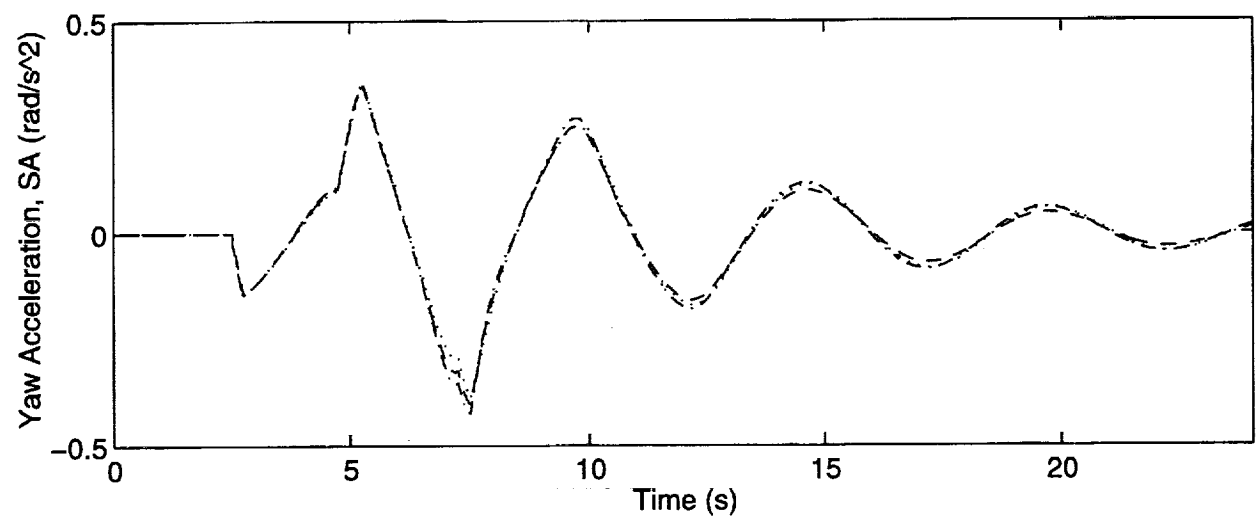
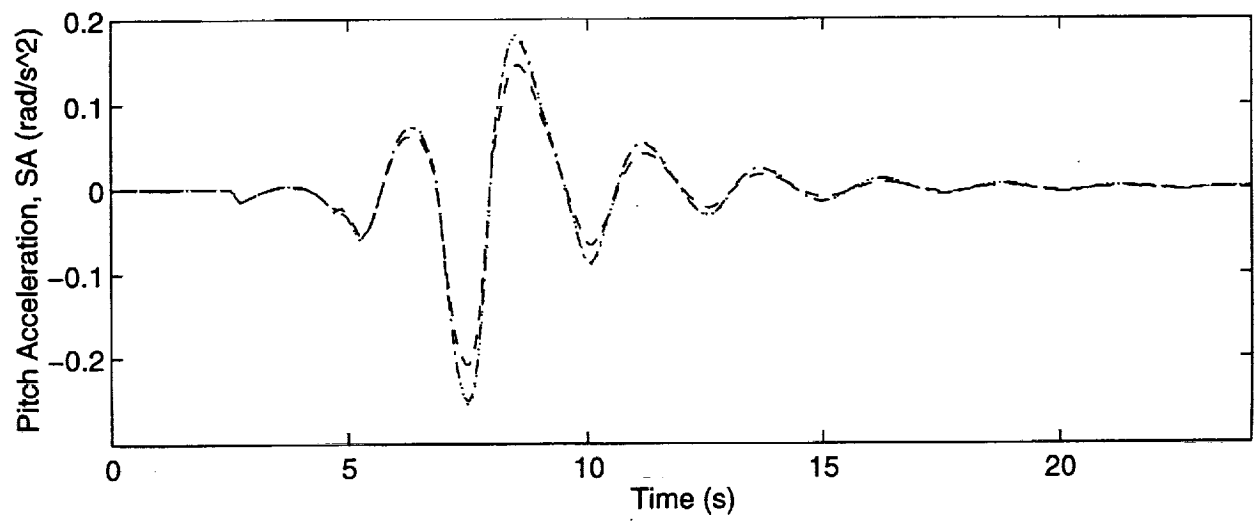
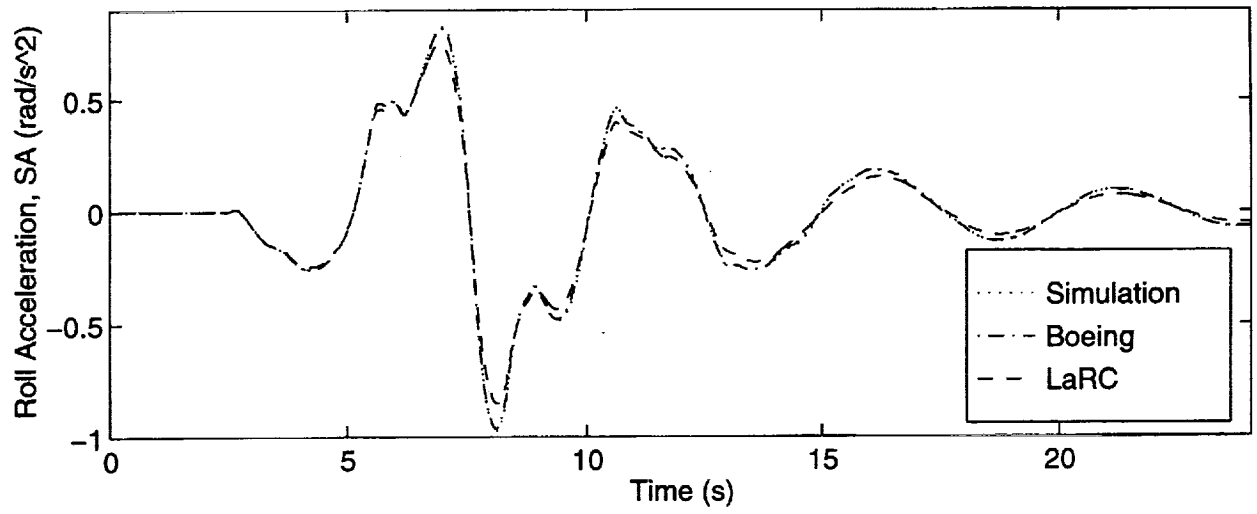
Nonlinear simulation results are compared with results from Boeing Commercial Aircraft Company and NASA Langley Research Center for the flight condition defined below. A 10 degree doublet is applied at all three rudder segments. The simulation was run at 40 Hz using an Euler integration algorithm.

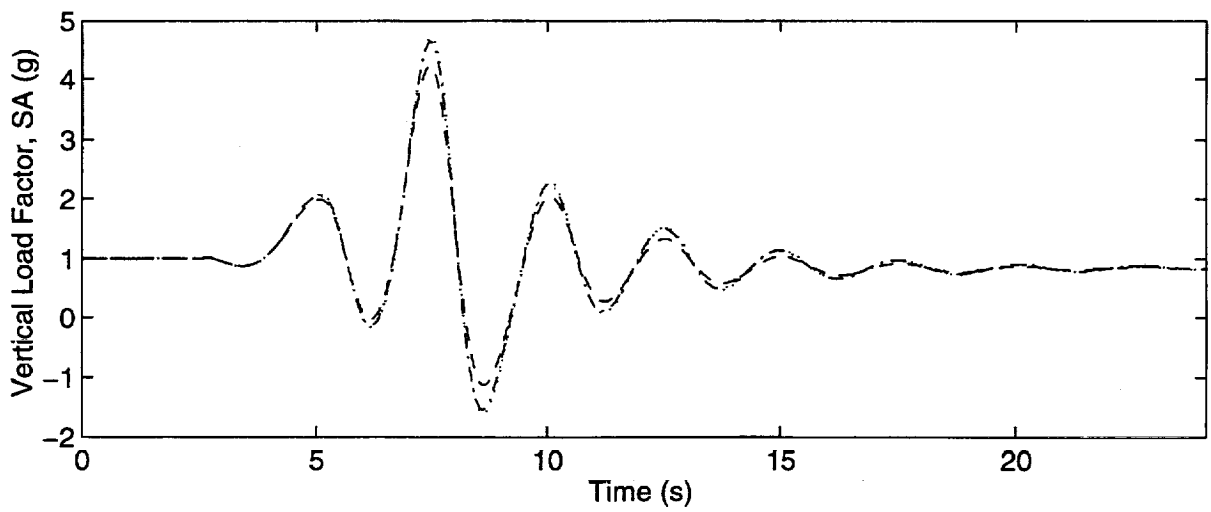
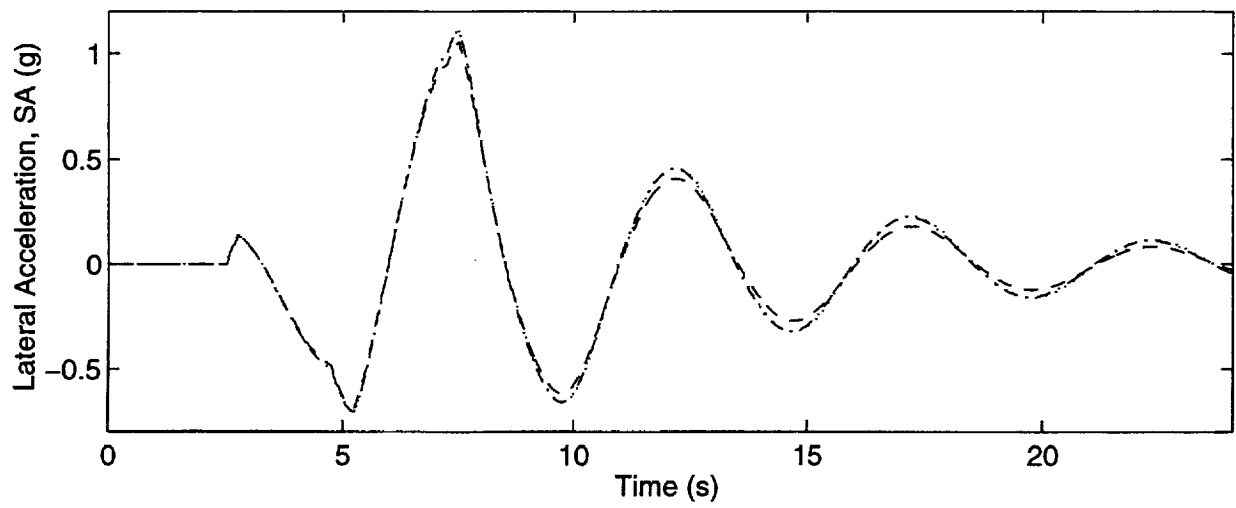
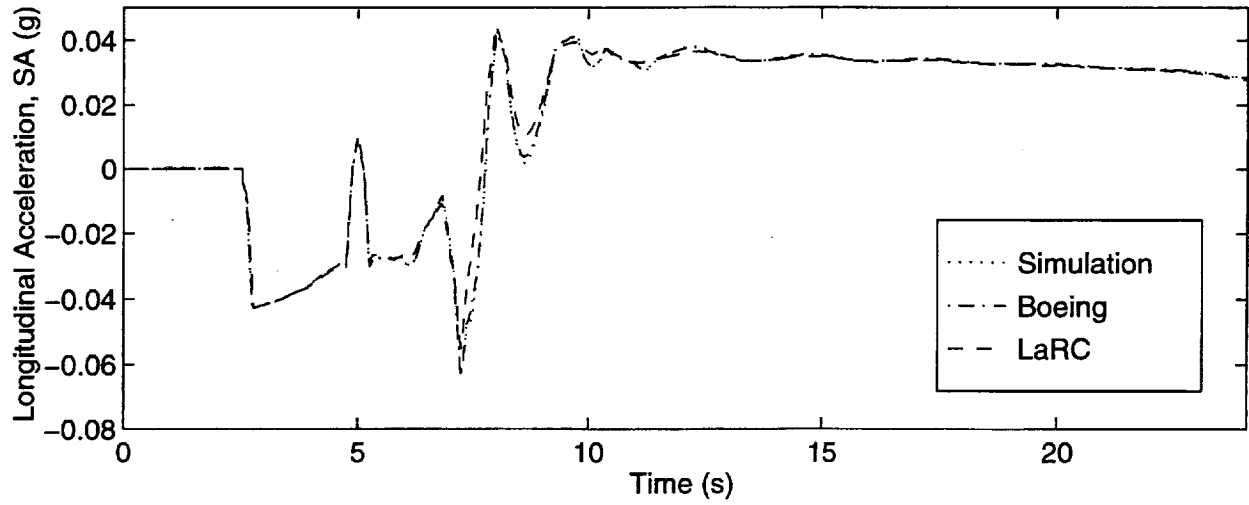
$u = 1036.57435$	$p = 0.$	$h = 20000.$	$\phi = -1.32394\text{e-}23$
$v = 0.$	$q = 0.$	$\lambda = 0.$	$\theta = 0.0261307$
$w = 27.092043$	$r = 0.$	$\tau = 0.$	$\psi = 5.58502\text{e-}06$
DLE1 = 10.	DTE1 = 3.	DTE5 = 0.	PLA1_CP = 68.4916
DLE2 = 0.	DTE2 = 3.	DTE6 = 0.	PLA2_CP = 68.4916
DLE3 = 0.	DTE3 = 0.	DTE7 = 3.	PLA3_CP = 68.4916
DLE4 = 10.	DTE4 = 0.	DTE8 = 3.	PLA4_CP = 68.4916
DRUD1 = 0.	DVF = 0.	DSTAB = -0.426949	
DRUD2 = 0.	DGEAR = 0.	DELEV = -0.853989	
DRUD3 = 0.			
GW = 279080.			
XCG_PMAC = 54.602			
YCG_BL_MD = 0.			
ZCG_WL_MD = 201.5			
INIT_FLAG = 0			
RIGID_FLAG = 0			
ROTATION_FLAG = 0			
DRYDEN_FLAG = 0			
TLIMIT_FLAG = 1			
AUTO CG_FLAG = 1			
AUTOWT_FLAG = 1			

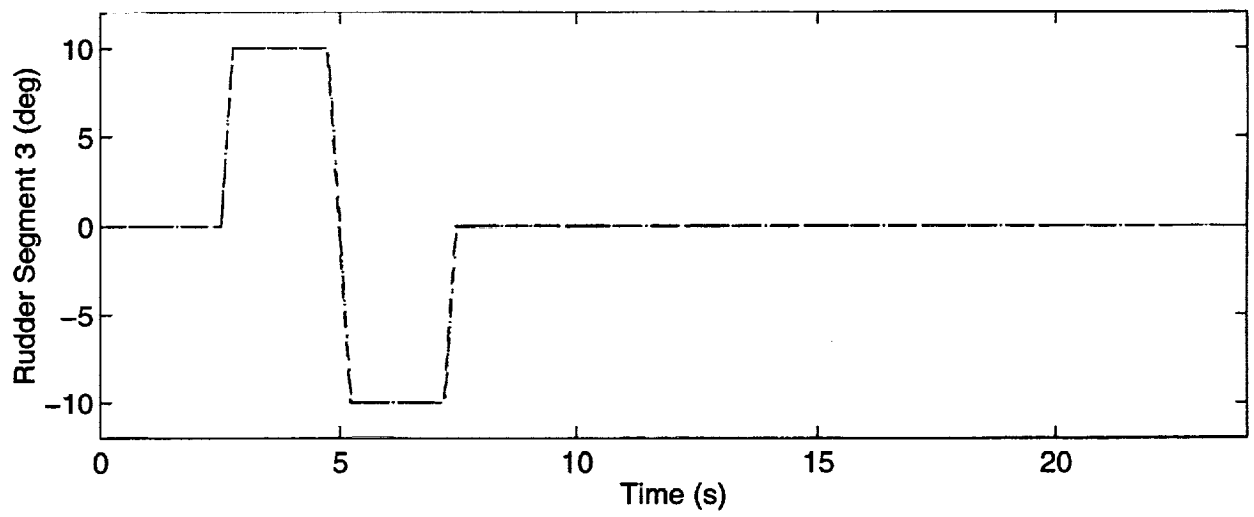
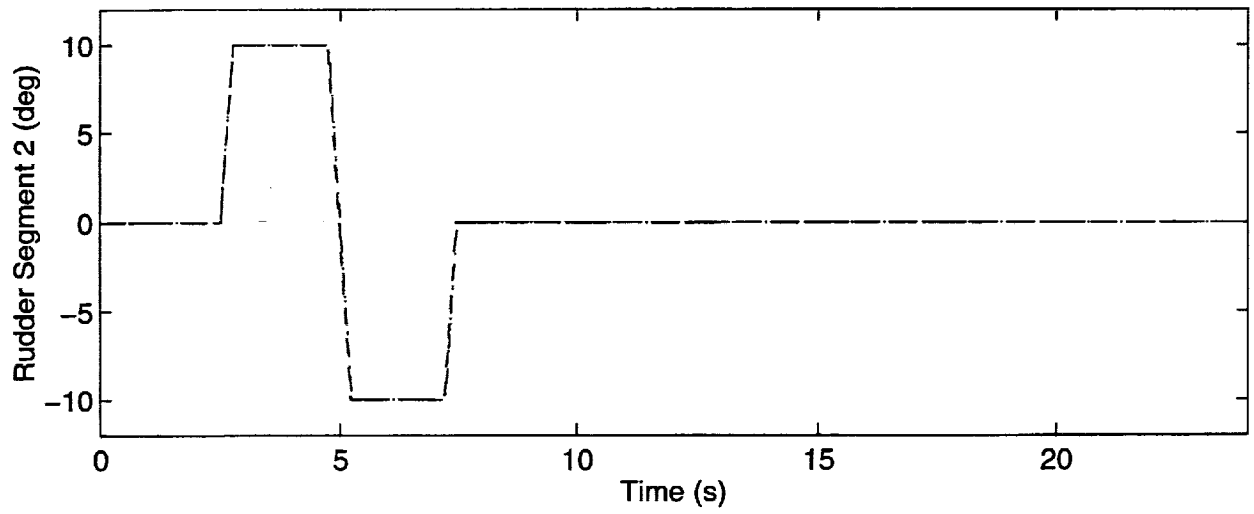
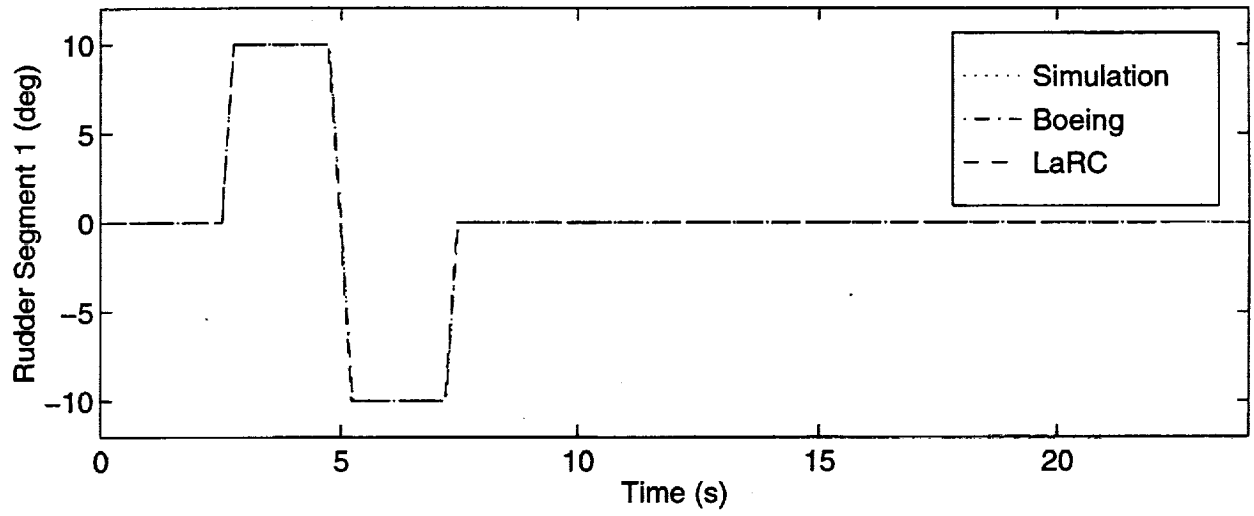












REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1999	3. REPORT TYPE AND DATES COVERED Technical Memorandum		
4. TITLE AND SUBTITLE High Speed Civil Transport Aircraft Simulation: Reference-H Cycle 1 MATLAB Implementation		5. FUNDING NUMBERS WU 537-07-24-01		
6. AUTHOR(S) Robert A. Sotack, Rajiv S. Chowdhry, and Carey S. Buttrill				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199		8. PERFORMING ORGANIZATION REPORT NUMBER L-17900		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA/TM-1999-209530		
11. SUPPLEMENTARY NOTES Sotack: George Washington Univ., JIAFS, Langley Research Center, Hampton, VA; Chowdhry: Lockheed Martin Engineering & Sciences Co., Hampton, VA; Buttrill: Langley Research Center, Hampton, VA.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 08 Distribution: Nonstandard Availability: NASA CASI (301) 621-0390		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) The mathematical model and associated code to simulate a high speed civil transport aircraft - the Boeing Reference H configuration - are described. The simulation was constructed in support of advanced control law research. In addition to providing time histories of the dynamic response, the code includes the capabilities for calculating trim solutions and for generating linear models. The simulation relies on the nonlinear, six-degree-of-freedom equations which govern the motion of a rigid aircraft in atmospheric flight. The 1962 Standard Atmosphere Tables are used along with a turbulence model to simulate the Earth atmosphere. The aircraft model has three parts - an aerodynamic model, an engine model, and a mass model. These models use the data from the Boeing Reference H cycle 1 simulation data base. Models for the actuator dynamics, landing gear, and flight control system are not included in this aircraft model. Dynamic responses generated by the nonlinear simulation are presented and compared with results generated from alternate simulations at Boeing Commercial Aircraft Company and NASA Langley Research Center. Also, dynamic responses generated using linear models are presented and compared with dynamic responses generated using the nonlinear simulation.				
14. SUBJECT TERMS High Speed Civil Transport aircraft; Flight simulation; MATLAB, Equations of motion; Flight dynamics		15. NUMBER OF PAGES 81		16. PRICE CODE A05
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102